TICSP Series #19

Adaptive Varying Scale Methods in Image Processing

Part I: Denoising and Deblurring

Vladimir Katkovnik, Karen Egiazarian
Jaakko Astola

TICSP Series

1. Egiazarian, Saramäki, Astola. Proceedings of Workshop on Transforms and Filter Banks.
2. Yaroslavsky. Target Location: Accuracy, Reliability and Optimal Adaptive Filters.
3. Astola. Contributions to Workshop on Trends and Important Challenges in Signal Processing.
4. Creutzburg, Astola. Proceedings of The Second International Workshop on Transforms and Filter Banks.
5. Stanković, Moraga, Astola. Readings in Fourier Analysis on Finite Non-Abelian Groups.
6. Yaroslavsky. Advanced Image Processing Lab.: An educational and research package for MatLab.
7. Klapuri. Contributions to Technical Seminar on Content Analysis of Music and Audio.
8. Stanković, Stanković, Astola, Egiazarian. Fibonacci Decision Diagrams.
9. Yaroslavsky, Egiazarian, Astola. Transform Domain Image Restoration Methods: Review, Comparison and Interpretation.
10. Creutzburg, Egiazarian. Spectral Techniques and Logic Design for Future Digital Systems. Proceedings of International Workshop SPECLOG'2000.
11. Katkovnik. Adaptive Robust Array Signal Processing for Moving Sources and Impulse Noise Environment (Nonparametric M -estimation Approach).
12. Danielian. Regularly Varying Functions, Part I: Criteria and Representations.
13. Egiazarian, Saramäki, Astola. Proceedings of International Workshop on Spectral Methods and Multirate Signal Processing, SMMSP'2001.
14. Stankovic, Sasao, Astola. Publications in the First Twenty Years of Switching Theory and Logic Design.
15. Saramäki, Yli-kaakinen. Design of Digital Filters and Filter Banks by Optimization: Applications.
16. Danielian. Optimization of Functionals on Classes of Distributions with Moment's Constraints, Part I: Linear Case.
17. Egiazarian, Saramäki, Astola. Proceedings of Second International TICSP Workshop on Spectral Methods and Multirate Signal Processing, SMMSP'2002.
18. Danielian. Optimization of Functionals on Classes of Distributions with Moment's Constraints, Part II: Nonlinear Case.
19. Katkovnik, Egiazarian, Astola. Adaptive Varying Scale Methods in Image Processing. Part I: Denoising and Deblurring.

Tampere International Center for Signal Processing
Tampere University of Technology
P.O. Box 553 , FIN-33101 Tampere, Finland

ISBN 952-15-0999-6
ISSN 1456-2774

TTY, Monistamo
2003

PREFACE

Digital images are produced by a variety of physical devices, including still and video cameras, x-ray devices, electron microscopes, radar, and ultrasound, and used for a variety of purposes, including entertainment, medical, business (e.g. documents), industrial, military, civil (e.g. traffic), security, and scientific. In today's computer networks, the amount of circulated digital images increases rapidly and enormously. However, images maybe distorted through different types of natural, say blurring, changing in illumination, movement, and processing causes, such as histogram equalization, quantization, smoothing, compression, noise corruption, geometric transformation. It is imperative in each case for an observer, human or machine, to extract useful information about the scene being imaged. Often the raw image is not directly suitable for this purpose and must be processed in some way. Such processing is called image enhancement or/and image reconstruction.

Roughly speaking the following problems are basic in image reconstruction techniques:

- Denoising;
- Edge detection and segmentation;
- Interpolation;
- Deblurring.

Image denoising is applied when an image degradation can be interpreted (modelled) as an additive usually random noise. This noise can be originated in physical properties of vision and registration devices as well as in exploited signal processing. Denoising means image filtering with a major intention separate the noise and the image.

Edge detection concerns the localization of significant variations of the image intensity and the identification phenomenon that originated them. The problem and the definition of what is known as edge detection have been extensively studied and analyzed, yielding an overwhelmingly amount of solutions. The problem is so important due to the one of the fundamental properties of the human visual system which automatically and firstly recognizes edges, and only after that can discriminate objects from background, creates assumptions about the physical composition of the elements in a scene and their direct and subjective interrelations. *Segmentation* means a separation of the image in a number of sub-images usually blundered by edges. It can be different in a number of aspects and used for image enhancement as well as for further image interpretation.

Image *interpolation* has many applications in digital image processing. It addresses the problem of generating a high-resolution image from its low resolution version. Image resizing, rotation, scaling all these operation require the interpolation of image intensity function. The interpolation is widely used in the medical imaging as well as in digital photography especially as consumer digital photography is becoming ever more popular. From enlarging consumer images to creating large artistic prints, interpolation is at the heart of it all.

Image *deconvolution* refers to the problem of recovering an image from its blurred and noisy observations. Focusing and auto-focusing are problems from this field. They are of a great importance in many scientific, technical and consumer areas. Under the names deblurring, blind deblurring, as well as deconvolution, blind deconvolution, or in more general terms as the inverse problem all these problems are studied during decades. While there is a definite progress in this direction the problem continues to be quite challenging in particular because it belongs to the class of the so-called ill-posed problems.

There is an extensive literature on the digital image processing covering a huge world of different concepts, mathematical models and methods developed for the above mentioned signal processing problems.

This book is devoted to quite a recent original approach to the image processing based on two fruitful and independent ideas: the local polynomial approximation (LPA) and the intersection of confidence intervals (ICI) rule. This new approach and developed techniques are applicable to all above mentioned signal processing problems.

The LPA is a nonparametric technique which is applied for a linear filter design using a parametric polynomial data fit in a sliding window. The window size of this fit is one of the key-parameters of the LPA estimator. It is shown that the window size of the LPA can be interpreted as a scale parameter of estimation. In what follows we use the terms "window size" and "scale" as interchangeable. The LPA nonparametric estimators are well known and popular in statistics and signal processing. However, they never have been applied and studied in detail for image processing problems.

The LPA estimators are combined with an adaptive data-driven scale procedure called the ICI rule. The idea of this ICI adaptation is simple. The algorithm searches for a largest local vicinity of the point of estimation where the LPA assumption fits well to the data. The estimates of the image are calculated for a grid of window sizes and compared. The adaptive window size is defined as the largest of those windows in the grid for which the estimate does not differ significantly from the estimators corresponding to the smaller window sizes. Overall the adaptive estimator is always nonlinear even for the linear LPA filter as the non-linearity of the method is incorporated by an ICI adaptive choice of the window size. It is proved that the ICI adaptive LPA filters are allowing to get a near optimal quality of image and edge recovery.

The $LPA - ICI$ technique is a powerful tool for image processing and first of all for denoising and differentiation. Let us remind that derivative estimation is a starting key-point of many edge detection procedures.

A further development of the LPA and the ICI produced for interpolation and deconvolution problems demonstrates that the scale adaptive LPA gives an strong and efficient background for all basic digital image processing problems.

The principal objectives of the book are to provide an introduction to basic concepts, methodology and theory of the LPA filter design and the ICI adaptive varying scale selection. In order to achieve these objectives, we focused on material that we believe is fundamental and has a scope of applications that is not limited to solutions of specialized problems.

The mathematical complexity of the book remains at a level well within the grasp of college seniors and first-year graduate students, who have introductory preparation in mathematical analysis, vectors, matrices, probability, statistics, and rudimentary computer programming.

Chapters I and II are focussed on the fundamentals of the LPA and the ICI statistics. The basic ideas are introduced and explained in details. The properties of the LPA estimates are given for discrete as well as integral versions. The filtering ability of the $LPA - ICI$ adaptive estimates are demonstrated on examples of image denoising.

The ICI adaptive estimation allows to introduce a new concept of the edge of image. The point is that the ICI rule gives a very small adaptive scale in the area where image intensity (or/and its derivatives) has a rapidly varying curvature, discontinuity or singularity. As a result the varying adaptive window sizes given as an image delineate very accurately all essential variations in the image. Thus, the adaptive varying scale can be used as an indicator of some singularities in image and applied as the corresponding edge detector.

Multiple denoising experiments are discussed in Chapter III.

In Chapter IV we introduce the ICI scale adaptive median estimates. It illustrates one of the important points concerning the used adaptation technique. The ICI based adaptation is quite universal and can be exploited with a number of quite different estimates not restricted to the class of the LPA estimators.

The deblurring is a subject of Chapter V. The proposed algorithm is based on direct pointwise estimation of the image intensity function. The estimator is derived from the LPA of the image in the adaptive size sliding window. The non-linearity of this method is incorporated by the ICI adaptive choice of the scale allowing to get a near optimal quality of image and edge recovery. The accuracy analysis shows that this estimator possesses the best possible ability for the pointwise adaptive estimation.

This book is a reflection of the significant progress, which is achieved in this field of the image processing using the proposed nonparametric *LPA* estimates and the *ICI* rule as independent tools as well as in a combination with more traditional methods and algorithms.

The concepts of the local approximation and the adaptive varying window are quite of universal nature. In particular, basis functions, different from polynomials, can be used for the local fitting and, in this way, the estimators can be yield which are different from the *LPA* and have different properties. In its turn the *ICI* statistics can be applied with estimators different from the *LPA*, say with sine/cosine transforms, wavelets, etc. It can be efficient for any methods using the series data fit of image segments when the size of the segment can be a subject of optimization.

The approach and methods presented in this book have natural links with the mathematical statistics, namely, the adaptive nonparametric estimation methods. In recent years a significant progress can be seen in this area of the mathematical statistics concerning both new ideas, methods and new technical mathematical achievements. In this book we try to minimize a necessary mathematical stuff to the level, which is sufficient for understanding of motivations and proposed algorithms. Nevertheless, the extensive list of the relevant statistical publications is included in Bibliography.

ACKNOWLEDGMENT

This work has been carried out mainly during March-September 2002 in Tampere International Center for Signal Processing (TICSP) of Tampere University of Technology when Vladimir Katkovnik was with TICSP as a Visiting Professor.

Tampere, Finland

April 2003

Contents

Chapter 1

LOCAL POLYNOMIAL APPROXIMATION

1.1 Introduction: image denoising

One typical problem of image analysis is a reconstruction of image from noisy data. It has been studied intensively within the last years (e.g. see books [107], [111], [7], [75]). Linear filtering techniques have been used in many image processing applications. The popularity of the linear methods stems from their mathematical simplicity and their efficiency in the presence of additive Gaussian noise. Linear filters, however, tend to blur sharp edges, destroy lines and other fine image details. This led to a search for nonlinear filtering alternatives. In particular, the development of nonlinear median-based filters in recent years has resulted in promising results and has highlighted some new promising research avenues. On account of its simplicity, edge preservation property, and robustness to impulsive noise, the standard median filter remains the quite popular for image processing applications [3]. The median filter, however, often tends to remove fine details in the image, such as thin lines and corners [3]. In recent years, a variety of median-type filters such as stack filters [93], multistage median [2], weighted median [94], and relaxed median [42] have been developed to overcome this drawback.

In many cases the image can be modelled by two dimensional ($2D$) function $y(x)$, $x \in R^2$, often composed of several regions with rather sharp edges. Within each region the image preserves a certain degree of uniformity while on the boundaries between the regions it has considerable changes. This leads to the edge estimation problem.

A large variety of methods has been proposed in the mathematical statistics for solving the image and edge estimation problem in different nonparametric context. The most popular methods of image estimation are based on the Bayesian or Markov random field approach, see [43], [26], [110], [124] among others. Nonparametric methods based on penalization and regularization have been developed in [118], [87] and [31]. Edge detection methods mostly do not assume any underlying parametric model. Methods based on kernel smoothing with a special choice of kernels have been discussed in [107], [77], [49], [86].

There is a number of proposals for nonparametric smoothing of images, which allow for preserving the sharp edge structure. We mention modal regression, see e.g. [113], the nonlinear Gaussian filter, see [33], the M-smoother [29], the adaptive weights smoother [104] and different proposals based on wavelets, see e.g. [98], [19] and [16] with references there.

Adaptation is now commonly considered as a crucial element of curve estimation procedures, in particular in nonparametric statistical methods. These adaptation methods are finalized in nonlinear estimators even for an originally linear method. The adaptation is a way to design nonlinear filters, which are at least in their derivation and initial motivation, are quite different from the mentioned above nonlinear filters mainly defined by the statistical models of noisy data. The literature on adaptive estimation suggests various methods, starting from classical cross-validation, C_p criterion and more recent techniques such as wavelet shrinkage [15] and the method of Lepski [78], [80], [81], which mainly we follow for in this book. A more complete review of the existing approaches and further references can be found in [15], [82], [46], [97], [5], [121], [105].

The adaptation used in this book is based on the direct nonparametric pointwise image estimation without any preliminary edge recovering. A linear estimator is applied, which is derived from the local polynomial approximation (LPA) of the image in a window selected by a data-driven way. The non-linearity of the method is incorporated by an adaptive choice of an estimation window allowing to get a near optimal quality of image and edge recovering. The idea of the used adaptation method is simple. The algorithm searches for a largest local vicinity of the point of estimation where the LPA assumption fits well to the data. The estimates of the image are calculated for a grid of window sizes and compared. The adaptive window size is defined as the largest of those windows in the grid, which the estimate does not differ significantly from the estimators corresponding to the smaller window sizes. This idea is common for Lepski's approach on the whole. However, the statistics used for comparison of the estimates of different window sizes are defined differently by different authors.

The presented approach can be viewed as one more application of the idea of pointwise adaptive estimation, see [78], [81], [82], [114]. The first three mentioned papers consider the problem of an adaptive choice of $1D$ estimator from a family of estimators which can be ordered by their variances. A typical example is given by kernel estimators with a fixed kernel and different window sizes. [114] discussed an adaptive choice of an asymmetric averaging window for local polynomial estimation including different nonsymmetric one-sided windows. The corresponding family is only partially ordered (i.e. there could be many estimators with the same or similar variance) and the original idea from Lepski [78] does not apply.

Further this idea have been applied to image estimation [105]. It is assumed that the image is composed of a finite number of regions and the image value is constant within each region. The number of regions, the difference between values of the image function for different regions and the regularity of edges are unknown and may be different for different parts of the image.

Among other approaches to this problem the LPA can be treated as probably one of the most theoretically justified and well studied ones. Originally LPA was proposed and developed in statistics for processing scalar and multidimensional noisy data. It is a powerful nonparametric technique, which provides estimates in a point-wise manner based on a mean square polynomial fitting in a sliding window (e.g. [55], [56], [58], [57], [14], [20], [46], [47], [59], [83]). In terms of the signal and image processing the LPA is a flexible tool to design $2D$ transforms having prescribed reproductive properties with respect to polynomial (smooth) components of signals. The invariant and variant optimal scale selection have been studied thoroughly by many authors. These optimal, in particular, varying data-driven scale methods are of special interest for the problems where the piece-wise smooth approximations are the most natural and relevant ones. Some image denoising problems provide good examples of these cases.

A crucial difference between the nonparametric LPA estimates and the more traditional parametric ones, say the polynomial mean squared estimates, is that the latter are formed as unbiased ones while the nonparametric former are biased on the definition and the reasonable choice of the biasedness controlled by the value of the scale is of importance. It can be emphasized that the problem of the optimal scale selection admits an accurate mathematical formulation in terms of the nonparametric approach, where the optimal scale is defined by a compromise between the bias and the variance of estimation.

Two main ideas are exploited for adaptive (data-driven) scale selection. The first one is based on estimation of the biasedness and the variance of the estimates of the signal with the corresponding optimal scale calculation based on theoretical formulas. However, the bias depends on the derivatives of a given signal. Thus, we need to estimate these derivatives and for this purpose to select auxiliary window sizes. Actually these methods, known as "pilot estimates", are quite complex in implementation and have quite a few design parameters.

The second alternative idea does not have to deal with the bias of estimation. This group of methods is based on the quality-of-fit statistics such as the cross-validation, generalized cross-validation, C_p , Akaike criteria, etc., which are applied for direct optimization of the accuracy.

A number of publications concerning the scale selection problem is very large and growing quickly. A review on the field or even its brief analysis is far beyond of the goal of this chapter. Here we give only a few references illustrating the basic progress in different directions.

A successful implementation of the first approach based on the pilot estimates has been reported by several authors. An automatic local scale selector with estimation of the higher order derivatives of $y(x)$, which are plugged into the local risk expression, was developed in [20]. The empirical-bias scale selection [112] uses the estimates at several window sizes in order to approximate the bias, and results in quite an efficient adaptive smoother for estimation of the function and its derivative. The similar ideas have been exploited in the adaptive smoothers described in [91].

Most of publications concerning the second approach are related to a data-based global (constant) scale selection (e.g. [47], [53], [59]). The linear LPA with the varying scale, found by minimization "pseudo-mean squared error", is considered in [85]. The target point is left out of the averaging in the pseudo-mean squared error what differs this method from the standard mean square methods. It is reported that the proposed pseudo-mean squared error works better then the local cross-validation [85].

This book exploits a quite recent new development. The intersection of confidence intervals (ICI) rule originally was proposed and developed in [37], [38] and [60] for denoising of $1D$ observations and shown to be quite efficient in particular for many distinct applications (e.g. [71], [72], [69]).

A generalization to two dimensional signals is proved to be efficient for gray-scale image denoising. In particular, in [63] and [62]. the image denoising problem is considered for quite a general observation model including image dependent noise. The denoising and scale selection are combined into one algorithm where the adaptive windows and signal denoising are produced on the basis of the same observation model.

In this chapter we present fundamentals, algorithms and analysis of the LPA method.

1.2 Observation model

The following model, commonly used for image processing, is assumed:

$$z(x_s) = y(x_s) + \varepsilon(x_s),\ s = 1, ..., n, \tag{1.1}$$

where an intensity $y(x)$ of the underlying image is defined as a function of two variables, $y \in R^1$, $x \in R^2$, $\varepsilon(x_s)$ is an additive noise, and n is a number of observations. It is assumed that all functions in (1.1) are defined on $2D$ rectangular regular grid with pixel's (grid node's) coordinates

$$x_s = (x_{1,s}, x_{2,s}).$$

In this notation image pixels are numbered by s taking values from 1 through n. The coordinates of the pixels can be given also explicitly:

$$\begin{aligned} &x_s = (k_1\Delta_1, k_2\Delta_2),\ k_1 = 1, ..., n_1, k_2 = 1, ..., n_2, \\ &n = n_1 n_2,\ \ s = 1, ...n, \end{aligned} \tag{1.2}$$

where Δ_1, Δ_2 are sampling intervals on the coordinates $x_{1,s}$ and $x_{2,s}$, respectively.

The noise is assumed to be random Gaussian, zero mean independent for different s with $E\{\varepsilon(x_s)\} = 0$, $E\{\varepsilon^2(x_s)\} = \sigma^2$.

The basic aim is to reconstruct (estimate) $y(x)$ from the observations $\{z(x_s)\}$ with the point-wise mean squared error (MSE) risk which is as small as possible.

It is assumed that $y(x)$ is unknown deterministic. In comparison with stochastic models of $y(x)$ it means that the main intention is to obtain the best result for every realization of $y(x)$ even if they are generated by a probabilistic phenomenon.

Different hypotheses on $y(x)$ can be applied for derivation and analysis of image processing algorithms. Here we follow the nonparametric approach assuming that a parametric representation of $y(x)$, say in the form of a series or function with reasonably small number of parameters, does not exist or unknown. Thus, $y(x)$ can be quite arbitrary, say with discontinuities and varying curvature. However, a piecewise smoothness of $y(x)$ is assumed, i.e. the grid, where $y(x)$ is defined, can be separated into M regions A_m, $m = 1, ..., M$, each of them is a connected set with an edge (boundary G_m). The function $y(x)$ is assumed to be smooth differentiable within each region A_m:

$$y(x) = \sum_{m=1}^{M} y_m(x)\mathbf{1}[x \in A_m], \tag{1.3}$$

where $\mathbf{1}[x \in A_m]$ is an indicator of the region A_m, $\mathbf{1}[x \in A_m] = 1$ if $x \in A_m$

and zero otherwise, and $y_m(x)$ is a smooth function belonging to a class of nonparametric continuous $r-$differentiable functions:

$$F_r(\bar{L}_r) = \{\max_{r_1+r_2=r} \left|D^{(r_1,r_2)}y(x)\right| = L_r(x) \leq \bar{L}_r,\ \forall\ r_1+r_2=r\}, \tag{1.4}$$

where $D^{(r_1,r_2)} = \frac{\partial^{r_1+r_2}}{\partial x_1^{r_1}\partial x_2^{r_2}}$ is a differentiation operator and $\bar{L}_r$ is finite invariant. The parameters r and $\bar{L}_r$ can be different for different $y_m(x)$ in (1.3).

The piecewise constant $y(x)$

$$y(x) = \sum_{m=1}^{M} a_m \mathbf{1}[x \in A_m], \tag{1.5}$$

is a particular case of (1.3) with constant values within each region A_m.

In the models (1.3) and (1.5) $y_m(x)$, a_m as well as the regions A_m are unknown. The boundaries G_m define change points of the piecewise smooth $y(x)$ in (1.3). The estimation of $y(x)$ can be produced in two different ways. One of the possible approaches deals with a two-stage procedure including estimation of boundaries G_m on the first stage, which defining the regions A_m. The second stage is a parametric or nonparametric fitting $y_m(x)$ on A_m. Another approach is connected with the concept of spatially adaptive estimation. In this context, change points or, more generally, cusps in the curves can be viewed as a sort of inhomogeneous behavior of the estimated function. One may therefore apply the same procedure, for instance nonlinear wavelet, ridgelet, curvlet estimators, and the analysis focuses on the quality estimation when change-points are incorporated in the model. Under this approach, the main intention is to estimate the function not location of change-points which are treated as elements of the function surface.

In this book we follow the second approach. The aim is to develop a method, which simultaneously adapts to inhomogeneous smoothness of the estimated curve and which is sensitive to discontinuities of the curves and its derivatives.

1.3 Local polynomial approximation

The idea of the LPA is simple and natural. It is assumed that the function $y(x)$ is well approximated by a polynomial in some neighborhood of the point of interest x. We find the coefficients of the polynomial fit by the weight least square method and use this approximation in order to calculate the estimate for the point of interest x called also "center" of the LPA. In fact, the local expansion is applied in order to calculate the estimate for this point of interest only. For the next point the calculations should be repeated. This pointwise procedure determines a nonparametric character of the LPA estimation.

1.3.1 Discrete LPA estimator

Let x be a "center" (desired point) of the LPA then the estimate for the point x_s in the neighborhood of the center x is looked for in the form

$$\begin{aligned}
&y(x, x_s) = C^T\phi(x - x_s), \qquad (1.6)\\
&\phi(x) = (\phi_1(x), \phi_2(x), ..., \phi_M(x))^T,\\
&C = (C_1, C_2, ..., C_M)^T,\ x = (x_1, x_2),\ x_s = (x_{1,s}, x_{2,s}),
\end{aligned}$$

where $\phi(x) \in R^M$ is a vector of linear independent $2D$ polynomials of the powers from 0 till m, $C \in R^M$ is a vector of parameters of this model. A total number of the polynomials and parameters is equal to

$$M = \frac{(m+2)!}{2 \cdot m!} = (m+2)(m+1)/2.$$

For $m = 3$ a complete set of the $2D$ linear independent polynomials can be given in the form

$$\begin{aligned}
&\phi_1 = 1,\ \text{for } m = 0; \qquad (1.7)\\
&\phi_2 = x_1,\ \phi_3 = x_2,\ \text{for } m = 1;\\
&\phi_4 = x_1^2/2,\ \phi_5 = x_2^2/2,\ \phi_6 = x_1x_2,\ \text{for } m = 2;\\
&\phi_7 = x_1^3/6,\ \phi_8 = x_2^3/6,\ \phi_9 = x_1^2x_2/2,\ \phi_{10} = x_1x_2^2/2,\ \text{for } m = 3;
\end{aligned}$$

with $M = \frac{(2+m)!}{2 \cdot m!} = 10$.

Note that the name center does not mean a central position of x in the neighborhood. Even more, x may not belong to the neighborhood. The center means that the LPA is applied in order to obtain the estimate of the function in this particular point.

The following criteria function is applied in the LPA in order find the coefficient C in (1.6) (e.g. [14], [20], [47], [59], [83]):

$$J_h(x) = \sum_s w_h(x - x_s)(z(x_s)) - y(x, x_s))^2, \tag{1.8}$$

where the window

$$w_h(x) = w(x/h)/h^2 \tag{1.9}$$

formalizes the localization of fitting with respect to the centre x, while the scale parameter $h > 0$ determines the size of the window. The window $w(x)$ is a function satisfying the conventional properties:

$$w(x) \geq 0, \; w(0) = \max_x w(x), \quad \int\int w(x_1, x_2) dx_1 dx_2 = 1. \tag{1.10}$$

A multiplicative window

$$w(x) = w_1(x_1) w_2(x_2), \tag{1.11}$$

where $w_1(x_1)$ and $w_2(x_2)$ are functions of scalar arguments, is commonly used. If the window is rectangular all observations enter in the criteria function with equal weights . Nonrectangular windows such as triangular, quadratic, Epanechnikov and so on (see [55], [56], [57], [14], [20], [47], [59], [83]) usually prescribe higher weights to observations which are closer to the centre x.

Here we can see some typical $1D$ window functions used in nonparametric estimates:

Table 1.1

NAME	$w(x)$
Rectangular symmetric	$1, \;\; \lvert x\rvert \leq 1/2,$
Rectangular nonsymmetric	$1, 0 \leq x \leq 1,$
Exponential	$\frac{1}{2}\exp(-\lvert x\rvert),$
Gaussian	$\frac{1}{\sqrt{2\pi}}\exp(-x^2/2),$
Epanechnikov	$\frac{3}{4}(1 - x^2), \;\; \lvert x\rvert \leq 1,$
Bisquare window	$(1 - x^2)^2, \lvert x\rvert \leq 1,$
Tricube	$(1 - \lvert x\rvert^3)^3, \;\; \lvert x\rvert \leq 1,$
Triweight window	$(1 - x^2)^3, \lvert x\rvert \leq 1.$

There is a simple way to generate nontrivial two-dimensional windows different from the multiplicative ones (1.11). Let us replace the argument x in the formulas given the above Table 1.1 by the norm $\lvert\lvert x\rvert\rvert$, where x is a vector and the norm is not exclusively Euclidian. Then after the corresponding normalization we obtain two-dimensional window functions satisfying (1.10).

Let $\hat{y}_h(x)$ be the LPA estimate of $y(x)$, where the subindex h shows a dependence of the estimate on the scale parameter. This LPA estimate of $y(x)$ is defined according to (1.6) as $\hat{y}_h(x) = y(x, x_s)|_{x_s=x} = y(x, x)$, i.e. the expansion (1.6) is used for estimate calculation for $x_s = x$ only. Actually, it is one of the key ideas of the pointwise nonparametric estimate design.

We introduce the estimate as some expansion in some neighborhood of the point x, we estimate the coefficients of this expansion, and finally we use this expansion only in order to estimate the function at this argument value x. Then, it follows from (1.6) that

$$\hat{y}_h(x) = y(x, x_s)|_{x_s=x} = C^T \phi(0) \tag{1.12}$$

and for the polynomials (1.7) it yields

$$\hat{y}_h(x) = y(x, x_s)|_{x_s=x} = C_1. \tag{1.13}$$

The LPA is applied also in order to design derivative estimators.

Let $\hat{y}_h^{(k)}(x)$ be the estimator of a $k - th$ derivative of $y(x)$. Here and in what follows

$$k = (k_1, k_2) \tag{1.14}$$

is a multi-index, k_1, k_2 nonnegative integer and $|k| = k_1 + k_2$. In particular, in this notation $x^k = x_1^{k_1} x_2^{k_2}$ and $\hat{y}_h^{(k)}(x)$ is the estimate of the derivative

$$\partial^{k_1+k_2} y(x)/\partial x_1^{k_1} \partial x_2^{k_2} = D^k y(x) = \partial^{|k|} y(x)/\partial x^k.$$

The LPA model (1.6) of the power m can be used for estimation of any derivative of the order k, $|k| \leq m$. According to the idea of the pointwise estimation we derive these estimates in the form

$$\hat{y}_h^{(k)}(x) = \frac{\partial y(x, x_s)|_{x_s=x}}{\partial x_{1,s}^{k_1} \partial x_{2,s}^{k_2}} = (-1)^{k_1+k_2} C^T \frac{\partial \phi(0)}{\partial x_{1,s}^{k_1} \partial x_{2,s}^{k_2}}. \tag{1.15}$$

This definition of the derivative estimator assumes that differentiation in (1.15) is done with respect to x_s as $y(x, x_s)$ is an approximation considered as a function of x_s provided that the LPA center x is fixed and after that we assume that $x_s = x$.

For the polynomials (1.7) the derivative estimates (1.15) are simple:

$$\begin{aligned} &\hat{y}_h^{(1,0)}(x) = -C_2,\ \hat{y}_h^{(0,1)}(x) = -C_3,\ \hat{y}_h^{(2,0)}(x) = C_4, \\ &\hat{y}_h^{(0,2)}(x) = C_5,\ \hat{y}_h^{(1,1)}(x) = C_6,\ \text{etc.} \end{aligned} \tag{1.16}$$

Thus, the coefficients of the LPA model (1.6) gives the estimates of the function and of the corresponding derivatives. Sometimes this clear link of the coefficients C with the function and derivative estimation is important for accurate formulation of the estimation problems.

The idea of the local approximation is applicable not only for the polynomials in the form (1.7) but also for different polynomials as well as for any basis functions $\phi(x)$ in (1.6) reasonable for the local fit. In this case, the estimates of the function and the derivatives are defined by the general formulas (1.12) and (1.15) and each estimate (function and derivative) can depend on all items of the vector C. The correspondence of the function and derivative estimates shown in (1.13)-(1.16) is valid only for the polynomials (1.7).

It deserves to be mentioned that (1.15) is not a unique definition of the derivative estimate. The estimate of the derivative $\partial^k y(x)/\partial x_1^{k_1}\partial x_2^{k_2}$ can be defined as the corresponding derivative of the function-estimate $\hat{y}_h(x)$. Then

$$\hat{y}_h^{(k)}(x) = \frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}}\hat{y}_h(x). \tag{1.17}$$

It can be verified that the estimates (1.15) and (1.17) are very different at least the derivatives of the window function $w(x)$ appear in (1.17) while do not appear in (1.15) [59].

Minimizing $J_h(x)$ (1.8) with respect to C,

$$\hat{C}(x,h) = \arg\min_{C\in R^M} J_h(x), \tag{1.18}$$

gives, provided $\det \Phi_h \neq 0$

$$\begin{aligned}
\hat{C}(x,h) &= \Phi_h^{-1}\sum_s w_h(x-x_s)\phi(x-x_s)z(x_s), && (1.19)\\
\Phi_h &= \sum_s w_h(x-x_s)\phi(x-x_s)\phi^T(x-x_s),\\
\hat{C}(x,h) &= (\hat{C}_1(x,h),...,\hat{C}_M(x,h))^T.
\end{aligned}$$

Substituting the coefficient estimates $\hat{C}(x,h)$ (1.19) in (1.13) and (1.15) instead of C we have the function and the derivative estimates in the form:

$$\hat{y}_h(x) = y(x,x) = \phi^T(0)\hat{C}(x,h), \tag{1.20}$$

$$\hat{y}_h^{(k)}(x) = (-1)^{|k|}[D^{(k_1,k_2)}\phi^T(0)]\cdot\hat{C}(x,h), \tag{1.21}$$

where

$$D^{(k_1,k_2)}\phi^T(0) = \frac{\partial}{\partial x_1^{k_1}\partial x_2^{k_2}}\phi^T(x)|_{x=0}.$$

The pointwise LPA estimates, as it is in (1.20) and (1.21), insure the reproduction properties of the estimate with respect to the polynomial components of $y(x)$.

But it should be emphasized that the LPA estimate of $y(x)$ is not a polynomial function. This is a principal difference between the nonparametric LPA and the corresponding parametric models.

For the polynomials (1.7)

$$\phi^T(0) = [1, 0..., 0, ..., 0] \tag{1.22}$$

is a zero vector-row $M \times 1$ with only $1st$ element equal to 1, and

$$[D^{(k_1,k_2)}\phi^T(0)] = [0, ..., 0, \underbrace{1}_{k_1,k_2-th}, 0, ..., 0] \tag{1.23}$$

is also a zero vector-row $M \times 1$ with the element corresponding to the polynomial $x_1^{k_1}x_2^{k_2}/k_1!k_2!$ equal to 1.

Then, we yield that $\hat{y}_h(x) \triangleq \hat{C}_1(x,h)$ is the estimate of the function, and $\hat{C}_l(x,h)$, $l = 2, ..., 10$, are the estimates of the derivatives of $y^{(1,0)}(x)$, $y^{(0,1)}(x)$, $y^{(2,0)}(x)$, $y^{(0,2)}(x)$, $y^{(1,1)}(x)$, etc., respectively, i.e.

$$\begin{aligned}
&\hat{y}_h(x) = \hat{C}_1(x,h),\ \hat{y}_h^{(1,0)}(x) = -\hat{C}_2(x,h), \hat{y}_h^{(0,1)}(x,h) = -\hat{C}_3(x,h),\\
&\hat{y}_h^{(2,0)}(x) = \hat{C}_4(x,h),\ \hat{y}_h^{(0,2)}(x,h) = \hat{C}_5(x,h), \hat{y}_h^{(1,1)}(x,h) = \hat{C}_6(x,h),\\
&\hat{y}_h^{(3,0)}(x,h) = -\hat{C}_7(x,h),\ \hat{y}_h^{(0,3)}(x,h) = -\hat{C}_8(x,h),\ \hat{y}_h^{(2,1)}(x,h) = -\hat{C}_9(x,h),\\
&\hat{y}_h^{(1,2)}(x,h) = -\hat{C}_{10}(x,h).
\end{aligned}$$

Thus, the estimate of the function is given only by the first item of the vector $\hat{C}(x,h)$ and other items of $\hat{C}(x,h)$ give the estimates of the derivatives ordered according the order of the $2D$ polynomials in the vector $\phi(x)$.

Substituting $\hat{C}(x,h)$ from (1.19) into (1.20) and (1.21) we obtain these estimates in the form of convolutions, i.e. linear filters

$$\begin{aligned}
\hat{y}_h^{(k)}(x,h) &= \sum_s g_h^{(k)}(x,x_s)z(x_s), \qquad (1.24)\\
g_h^{(k)}(x,x_s) &= w_h(x-x_s)[D^{(k)}\phi^T(0)]\Phi_h^{-1}\phi(x-x_s),\\
\Phi_h &= \sum_s w_h(x-x_s)\phi(x-x_s)\phi^T(x-x_s).
\end{aligned}$$

Here $g_h^{(k)}(x,x_s)$ is a kernel function of the differentiation $2D$ filter.

For the estimate of the function we use the above notation of the kernel function of the filter without indices k_1, k_2

$$\hat{y}_h(x) = \sum_s g_h(x,x_s)z(x_s), \tag{1.25}$$

where

$$g_h(x, x_s) = w_h(x - x_s)\phi^T(0)\Phi_h^{-1}\phi(x - x_s), \qquad (1.26)$$
$$\Phi_h = \sum_s w_h(x - x_s)\phi(x - x_s)\phi^T(x - x_s).$$

As the coefficients $\hat{C}$ are obtained by the least square methods it can be verified that

$$\sum_s g_h(x, x_s)\phi(x_s) = \phi(x), \qquad (1.27)$$
$$\sum_s g_h^{(k_1,k_2)}(x, x_s)\phi(x_s) = (-1)^{k_1+k_2} D^{(k_1,k_2)}\phi(x).$$

These conditions means that for any polynomials with power equal or less m, $y(x) \in P_m$, the estimates of the function and the derivatives are accurate:

$$\sum_s g_h(x, x_s)y(x_s) = y(x),$$
$$\sum_s g_h^{(k_1,k_2)}(x, x_s)y(x_s) = D^{(k_1,k_2)}y(x).$$

Thus, the filters designed according to the LPA method way give accurate values of the function and its derivatives (orders $k \leq m$) for any polynomials of the powers up to the value m.

The designed linear filters can be applied to data given on any regular or irregular grids, in particular, to data with lost observations and for data interpolation problem when the center x of the LPA does not belong to the regular grid. It is assumed in the above formulas that the summation is always performed within boundaries of the image frame.

Concerning the terminology we wish to note that in statistics the weights $g_h(x, x_s)$ and $g_h^{(k_1,k_2)}(x, x_s)$ are named "kernels" and the estimates (1.25) and (1.24) are "kernel estimates". The term "bandwidth" is used in statistics for the window-size (window-scale) parameter h. We will treat all these terms as the corresponding synonymies and apply all of them.

In image processing the term "mask" is used for the weights $g_h(x, x_s)$ and $g_h^{(k_1,k_2)}(x, x_s)$.

Further we mainly consider the case when the window function $w(x)$ in (1.8) has a finite support. For example $w(x) = 0$ if $||x|| = \sqrt{x_1^2 + x_2^2} > 1$. Then $w_h(x) = 0$ if $||x|| > h$. Thus, the parameter h defines a scale as well as a size of the support of the masks $g_h^{(k_1,k_2)}(x, x_s)$ and $g_h(x, x_s)$. A mask with a finite support is called finite and the corresponding filter is a finite impulse response (FIR) filter.

For a regular grid, when all x_s and x belong to the grid, and the image is unbounded (infinite) the weights $g_h(x, x_s)$ and $g_h^{(k_1,k_2)}(x, x_s)$ depend on the difference of the arguments x and x_s only

$$\begin{aligned}&g_h(x, x_s) = g_h(x - x_s)\\ &g_h^{(k_1,k_2)}(x, x_s) = g_h^{(k_1,k_2)}(x - x_s).\end{aligned}$$

This conclusion immediately follows from the formulas (1.26) (1.24). Then the estimates (1.24) and (1.25) can be represented in the form of the homogeneous (shift invariant) convolutions with the invariant on x (independent on x) weights:

$$\begin{aligned}&\hat{y}_h^{(k_1,k_2)}(x) = \sum_s g_h^{(k_1,k_2)}(u_s) z(x - u_s), \qquad (1.28)\\ &g_h^{(k_1,k_2)}(u_s) = w_h(u_s)[D^{(k_1,k_2)}\phi^T(0)]\Phi_h^{-1}\phi(u_s),\\ &\Phi_h = \sum_s w_h(u_s)\phi(u_s)\phi^T(u_s)\end{aligned}$$

and

$$\begin{aligned}&\hat{y}_h(x) = \sum_s g_h(u_s) z(x - u_s), \qquad (1.29)\\ &g_h(u_s) = w_h(u_s)\phi^T(0)\Phi_h^{-1}\phi(u_s).\end{aligned}$$

In this representation the variables

$$u_s = x - x_s$$

are relative coordinates of the grid nodes with respect to the estimation point x.

Practically, the nonhomogeneous and homogeneous weights are close to each other provided that the estimation points x are far enough from the image boundaries.

The reproduction properties (1.27) are simplified for the homogeneous weights to the form

$$\begin{aligned}&\sum_s g_h(u_s)\phi(u_s) = \phi(0), \qquad (1.30)\\ &\sum_s g_h^{(k_1,k_2)}(u_s)\phi(u_s) = (-1)^{k_1+k_2} D^{(k_1,k_2)}\phi(0).\end{aligned}$$

The linear estimators (1.20)-(1.21) and (1.28)-(1.29) have a very long prehistory (e.g. [57], [14], [20], [46], [47], [59], [83]). They are a very popular tool in statistics and signal processing with application to a wide variety of the fields for smoothing, filtering, differentiation, interpolation and extrapolation.

1.3.2 Nadaray-Watson estimators

Historically first nonparametric estimates of the kernel type have been proposed in 1964 by Nadaray [88] and Watson [123] based on quite different concepts.

The Nadaray estimator has been proposed in the form of the weighted sum of the observations

$$\hat{y}_h(x) = \frac{1}{\sum_s w_h(x-x_s)} \sum_s w_h(x-x_s) z(x_s), \tag{1.31}$$

where a nonnegative window $w_h(x-x_s)$ defines a measure of a distance between the observation x_s and the target point x.

These distances being normalized define the estimate (1.31) in the form (1.29)

$$\begin{aligned} \hat{y}_h(x) &= \sum_s g_h(x-x_s) z(x_s), \\ g_h(u) &= \frac{w_h(u)}{\sum_s w_h(u_s)},\ u_s = x - x_s, \end{aligned} \tag{1.32}$$

with the kernel $g_h(u)$.

The Watson estimator again in the form (1.31) has been derived (not just proposed heuristically) using statistical concepts.

Let x and z in (1.1) be random Gaussian. Then the conditional expectation of z gives y as

$$y(x) = E\{Z/X = x\} = \int z f(z,x) dz / \int f(z,x) dz, \tag{1.33}$$

where $E\{Z/X = x\}$ is the conditional expectation of the random Z provided that the random $X = x$. The integrals in (1.33), which can be multivariable, define corresponding calculations using the joint probability density $f(z,x)$ of the random Z and X.

Let $\hat{f}_h(z,x)$ be the Parzen kernel estimator of the probability density $f(z,x)$ [102]

$$\hat{f}_h(z,x) = \frac{1}{\# \cdot h^2} \sum_s \rho(\frac{z - z(x_s)}{h}, \frac{x - x_s}{h}), \tag{1.34}$$

$$\int \rho(z,x) dx dz = 1,\ \int z \rho(z,x) dz = 0.$$

The kernel $\rho \geq 0$ is even on the both arguments, $\rho(z,x) = \rho(-z,x)$, $\rho(z,x) = \rho(z,-x)$.

Substituting (1.34) in (1.33) gives (1.31) with

$$w(u) = \int \rho(x,u)dx,\ w_h(u) = \frac{1}{h}w(\frac{u}{h})$$

Thus, we arrive to the estimator (1.31) from quite a different starting point using the ideas of the statistically optimal estimation.

Now let us go to the Nadaray-Watson estimator from minimization of the quadratic lost function (1.8) and define the estimate as a solution of the problem

$$\hat{y}_h(x) = \hat{C}(x,h) = \arg\min_C(\sum w_h(x-x_s)[z(x_s)-C]^2). \tag{1.35}$$

It immediately gives the formula (1.31) for the estimate $\hat{y}_h(x)$.

Thus, we derive the same estimator (1.31) starting from three quite different concepts: the heuristic idea of Nadaray, the statistical estimator of Watson and the LPA estimation.

In the terms of the LPA the Nadaray-Watson estimator can be interpreted as the LPA with the power $m = 0$. The Nadaray-Watson estimator actually assumes that the estimated function is locally constant while in general we use a more general assumption that the estimated function can be fitted locally by some given functions $\phi_k(x)$, $k = 1, ..., M$, which, in particular, can be polynomials of some power m.

1.3.3 Constrained LPA estimation

Different hypotheses on $y(x)$ can be exploited in image processing. One of the most usual is an assumption that $y(x)$ is an arbitrary nonnegative function normalized to the interval $[0,1]$, i.e.

$$0 \leq y(x) \leq 1, \tag{1.36}$$

with possible discontinuities and varying curvature.

Let us discuss some issues concerning estimation of $y(x)$ subject to the constrain (1.36). This inequality can be naturally incorporated in (1.18) and result in the following constrained optimization problem for estimation of C:

$$\hat{C}(x,h) = \arg \min_{\substack{0\leq C_1 \leq 1 \\ \tilde{C}\in R^{M-1}}} J_h(x). \tag{1.37}$$

Here $\tilde{C} = (C_2, ..., C_M)^T$ is the vector C with the excluded first item C_1. The restriction (1.36) according to (1.13) is imposed only on C_1 while all other items of C continue to be unrestricted. When the estimates of the coefficients are found from (1.37) the intensity and the derivatives are calculated according to the formulas (1.13)-(1.15).

Thus, the constrained estimate of the image intensity always satisfy the conditions (1.36) while it cannot be guaranteed for the linear estimates (1.20) and (1.21) following from (1.18).

Concerning the estimates (1.37) we wish to note a number of interesting moments. First, in general, these estimates are nonlinear with respect to the observations and their calculation is a much more complex problem as compared with the case of the linear estimates and unconstrained optimization. However, if the linear estimate $\hat{C}_1(x,h)$ found from (1.18) belongs to the interval $[0,1]$ then the solutions of (1.18) and (1.37) are indentical. It gives a simple logic to deal with this nonlinear problem. We start from the calculation of the linear estimates for all x and test them on the restrictions (1.36). In this way we identify the pixels violating the conditions (1.36) and the nonlinear constrained estimates (1.37) are calculated only to these pixels.

Second, if the linear estimate does not satisfy the conditions (1.36) the solutions of the unconstrained and constrained optimizations can be quite different by all elements of the vector C not only by C_1. The constrained optimization (1.37) enables one to yield both the more accurate estimate of the image intensity as well as its derivatives. The constrain on the intensity function influences the estimates of the derivatives.

The following compromise can be used in order to avoid the complexity of the constrained problem (1.37). The estimates are obtained in two independent steps. The first step is a solution of the unconstrained problem ignoring the restrictions on C_1. The second step defines the estimates according to the equations:

$$\hat{y}_h(x) = [(g_h \circledast z)(x)]_+, \tag{1.38}$$

$$\hat{y}_h^{(k)}(x) = (g_h^{(k)} \circledast z)(x), \tag{1.39}$$

where $[\cdot]_+$ stays for the projection on the segment $[0,1]$, which means $[x]_+ = x$ for $0 \leq x \leq 1$, $[x]_+ = 0$ for $x < 0$ and $[x]_+ = 1$ for $x > 1$. Thus, the estimate of C_1 obtained from the unconstrained optimization is replaced by its projection on the interval $[0,1]$. All others elements of the vector C are assumed to be equal to the corresponding items of the vector $\hat{C}(x,h)$.

The estimate built in this way can be treated as an approximation of the accurate nonlinear constrained estimate obtained from (1.18).

Note that it is a conventional practice in image processing and image filter design to ignore the nonnegativity and upper bound of the intensity function. The LPA approach allows to involve the restrictions on $y(x)$ in quite a simple way with the optimization problem where the constrained imposed on one of the variables only.

It can be mentioned also that mainly the linear estimates leaving the segment $[0, 1]$ can happened only for estimation nearly the minimum/maximum bounds of $y(x)$ or at the discontinuous of $y(x)$ when the high-order $(m \geq 1)$ LPA is applied. In what follows for the sake of simplicity we use for the analysis the linear estimate given as the convolutions (1.29)-(1.29).

1.4 Regular grid estimators

Using regular grid's coordinates in the form (1.2) we can rewrite the homogeneous estimates (1.28) and (1.29) in the form of the double-sum $2D$ convolutions

$$\begin{aligned}
&\hat{y}_h(x) = \sum_{l_1}\sum_{l_2} g_h(l_1\Delta_1,\ l_2\Delta_2) z(x_1 - l_1\Delta_1,\ x_2 - l_2\Delta_2), \qquad (1.40)\\
&g_h(l_1\Delta_1,\ l_2\Delta_2) = w_h(l_1\Delta_1,\ l_2\Delta_2)\phi^T(0,0)\Phi_h^{-1}\phi(l_1\Delta_1,\ l_2\Delta_2),\\
&\Phi_h = \sum_{l_1}\sum_{l_2} w_h(l_1\Delta_1,\ l_2\Delta_2)\phi(l_1\Delta_1,\ l_2\Delta_2)\phi^T(l_1\Delta_1,\ l_2\Delta_2), \ (1.41)
\end{aligned}$$

and

$$\begin{aligned}
&\hat{y}_h^{(k)}(x) = \sum_{l_1}\sum_{l_2} g_h^{(k)}(l_1\Delta_1,\ l_2\Delta_2) z(x_1 - l_1\Delta_1,\ x_2 - l_2\Delta_2), \qquad (1.42)\\
&g_h^{(k)}(l_1\Delta_1,\ l_2\Delta_2) = w_h(l_1\Delta_1,\ l_2\Delta_2)[D^{(k)}\phi^T(0,0)]\Phi_h^{-1}\phi(l_1\Delta_1,\ l_2\Delta_2),\\
&\Phi_h = \sum_{l_1}\sum_{l_2} w_h(l_1\Delta_1,\ l_2\Delta_2)\phi(l_1\Delta_1,\ l_2\Delta_2)\phi^T(l_1\Delta_1,\ l_2\Delta_2), \qquad (1.43)
\end{aligned}$$

where k is a multiple index (1.14) and ϕ (1.6) is a vector of $2D$ variable functions

$$\phi(l_1\Delta_1,\ l_2\Delta_2) = [\phi_1(l_1\Delta_1,\ l_2\Delta_2),\ \phi_2(l_1\Delta_1,\ l_2\Delta_2), ..., \phi_M(l_1\Delta_1,\ l_2\Delta_2)]^T.$$

Using the standard notation for the $2D$ convolutions, the estimates (1.40)-(1.42) can be represented in the following compact form

$$\begin{aligned}
&\hat{y}_h(x) = (g_h \circledast z)(x), \qquad (1.44)\\
&\hat{y}_h^{(k)}(x) = (g_h^{(k)} \circledast z)(x).
\end{aligned}$$

A dependence of the weights g_h and $g_h^{(k)}$ on the scale h can be made explicit. It is convenient also to introduce different scale parameters h_1 and h_2 for the arguments x_1 and x_2 respectively. In the following transformations we assume that the polynomials $\phi(x)$ are defined according to (1.7).

The weights g_h of the estimates can be represented in the forms

$$\begin{aligned}
&g_h(l_1\Delta_1,\ l_2\Delta_2) = \frac{1}{h_1h_2}w(\frac{l_1\Delta_1}{h_1},\ \frac{l_2\Delta_2}{h_2})\phi^T(0,0)\Phi^{-1}\phi(l_1\Delta_1,\ l_2\Delta_2) =\\
&\frac{1}{h_1h_2}w(\frac{l_1\Delta_1}{h_1},\ \frac{l_2\Delta_2}{h_2})\phi^T(0,0)\Phi^{-1}\phi(h_1\frac{l_1\Delta_1}{h_1}, h_2\ \frac{l_2\Delta_2}{h_2}). \qquad (1.45)
\end{aligned}$$

Introduce a diagonal matrix $T_d(h_1, h_2)$ composed from the items of the vector $\phi(h_1,\ h_2)$

$$T_d(h_1, h_2) = diag\{\phi(h_1, h_2)\} \tag{1.46}$$

and the variables

$$n_1 = \frac{h_1}{\Delta_1},\ n_2 = \frac{h_2}{\Delta_2},$$

defining the scale calculated in number of samples in the window. These variables n_1, n_2 can be integer or non-integer.

Using (1.46) and variables n_1, n_2 we have

$$\phi(h_1\frac{l_1\Delta_1}{h_1}, h_2\ \frac{l_2\Delta_2}{h_2}) = T_d(h_1, h_2)\phi(\frac{l_1}{n_1},\ \frac{l_2}{n_2})$$

and $g_h(l_1\Delta_1,\ l_2\Delta_2)$ in (1.45) can be rewritten as as

$$\begin{aligned} &g_h(l_1\Delta_1,\ l_2\Delta_2) = \\ &\frac{1}{h_1h_2}w(\frac{l_1}{n_1},\ \frac{l_2}{n_2})\phi^T(0,0)\Phi^{-1}T_d(h_1, h_2)\phi(\frac{l_1}{n_1},\ \frac{l_2}{n_2}). \end{aligned} \tag{1.47}$$

In a similar way, the matrix Φ (1.43) is of the form

$$\begin{aligned} &\Phi_h = \\ &\frac{1}{h_1h_2}\sum_{l_1}\sum_{l_2} w(\frac{l_1\Delta_1}{h_1},\ \frac{l_2\Delta_2}{h_2})\phi(l_1\Delta_1,\ l_2\Delta_2)\phi^T(l_1\Delta_1,\ l_2\Delta_2) = \\ &\frac{1}{h_1h_2}T_d(h_1, h_2)[\sum_{l_1}\sum_{l_2} w(\frac{l_1}{n_1},\ \frac{l_2}{n_2})\phi(\frac{l_1}{n_1},\ \frac{l_2}{n_2})\phi^T(\frac{l_1}{n_1},\ \frac{l_2}{n_2})]T_d(h_1, h_2) = \\ &\frac{1}{h_1h_2}T_d(h_1, h_2)\Psi T_d(h_1, h_2) \end{aligned} \tag{1.48}$$

$$\Psi = \sum_{l_1}\sum_{l_2} w(\frac{l_1}{n_1},\ \frac{l_2}{n_2})\phi(\frac{l_1}{n_1},\ \frac{l_2}{n_2})\phi^T(\frac{l_1}{n_1},\ \frac{l_2}{n_2}) \tag{1.49}$$

Substituting (1.47) and (1.48) in (1.45) gives the weights $g_h(l_1\Delta_1, l_2\Delta_2)$ in a new form:

$$\begin{aligned} &g_h(l_1\Delta_1,\ l_2\Delta_2) = g(l_1, l_2), \\ &g(l_1, l_2) = w(\frac{l_1}{n_1},\ \frac{l_2}{n_2})\phi^T(0,0)\Psi^{-1}\phi(\frac{l_1}{n_1},\ \frac{l_2}{n_2}) \\ &\Psi = \sum_{l_1}\sum_{l_2} w(\frac{l_1}{n_1},\ \frac{l_2}{n_2})\phi(\frac{l_1}{n_1},\ \frac{l_2}{n_2})\phi^T(\frac{l_1}{n_1},\ \frac{l_2}{n_2}). \end{aligned} \tag{1.50}$$

Similar calculations for the differentiation kernel $g_h^{(k)}(l_1\Delta_1,\ l_2\Delta_2)$ in (1.42)

$$g_h^{(k)}(l_1\Delta_1,\ l_2\Delta_2) = g^{(k)}(l_1,l_2),$$
$$g^{(k)}(l_1,l_2) = (-1)^{|k|}w(\frac{l_1}{n_1},\ \frac{l_2}{n_2})[D^{(k_1,k_2)}\phi^T(0,0)]T_d^{-1}(h_1,h_2) \times$$
$$\Psi^{-1}\phi(\frac{l_1}{n_1},\ \frac{l_2}{n_2}) =$$
$$\frac{1}{h_1^{k_1}h_2^{k_2}}(-1)^{|k|}w(\frac{l_1}{n_1},\ \frac{l_2}{n_2})[D^{(k_1,k_2)}\phi^T(0,0)]\Psi^{-1}\phi(\frac{l_1}{n_1},\ \frac{l_2}{n_2}). \quad (1.51)$$

It is used in this derivation that

$$[D^{(k_1,k_2)}\phi^T(0,0)]T_d^{-1}(h_1,h_2) = \frac{1}{h_1^{k_1}h_2^{k_2}}D^{(k_1,k_2)}\phi^T(0,0).$$

Thus, we obtain the kernels of the estimates (1.44) in the new form (1.50)-(1.51) which is more convenient for analytical as well as for numerical calculations.

Let n_1 and n_2 be fixed then the smoothing kernel does not depends on the sampling interval and can be universally used for all sampling intervals.

For the differentiation kernel the similar situation is valid as the kernel (1.51) is a product of two factors. The first main factor-function defined as

$$(-1)^{k_1+k_2}w(\frac{l_1}{n_1},\ \frac{l_2}{n_2})[D^{(k_1,k_2)}\phi^T(0,0)]\Psi^{-1}\phi(\frac{l_1}{n_1},\ \frac{l_2}{n_2})$$

is again invariant to values of the sampling intervals. The second factor is a scalar $h_1^{-k_1}h_2^{-k_2} = (n_1\Delta_1)^{-k_1}(n_2\Delta_2)^{-k_2}$, a clear analytical function of the sampling interval values. It defines the corresponding normalization of the derivative estimates.

Actually the new introduced kernels use instead of the original scale parameter h_1 and h_2 the new scale parameters normalized by the sampling interval values n_1 and n_2. Thus, of course, the new kernels depend on the scale.

The dependence of the estimates on the scale is of importance and we emphasize this dependence by using h as a subscript of the kernels and the estimates. In what follows we will continue to use the subscript h even for the kernels and the estimates given in the new forms (1.50) and (1.51) just in order to emphasize that the estimator and estimates depends on this key-parameter.

1.4.1 Analytical regular grid kernels

The equations (1.40) and (1.42) define the regular grid estimates for a general case. If $\Delta_1 = \Delta_2 = 1$ the homogeneous kernels of these estimates coincide with (1.50) and (1.51) and can be given as:

$$g_h(l_1,\ l_2) = w_h(l_1,\ l_2)\phi^T(0,0)\Phi_h^{-1}\phi(l_1,\ l_2) \tag{1.52}$$

and

$$g_h^{(k)}(l_1,\ l_2) = (-1)^{|k|}w_h(l_1,\ l_2)[D^{(k)}\phi^T(0,0)]\Phi_h^{-1}\phi(l_1,\ l_2), \tag{1.53}$$

with

$$\Phi_h = \sum_{l_1}\sum_{l_2} w_h(l_1,\ l_2)\phi(l_1,\ l_2)\phi^T(l_1,\ l_2), \tag{1.54}$$

for smoothing and differentiation respectively.

Here we are going to consider some particular cases when formulas for the kernels can be obtained in order to demonstrate what kind of kernel functions we can have for these estimates.

Let start from the $1D$ case when the above formulas are simplified to

$$g_h(l_1) = w_h(l_1)\phi^T(0)\Phi_h^{-1}\phi(l_1) \tag{1.55}$$

and

$$g_h^{(k_1)}(l_1) = (-1)^{k_1}w_h(l_1)[D^{(k_1)}\phi^T(0)]\Phi_h^{-1}\phi(l_1), \tag{1.56}$$

$$\Phi_h = \sum_{l_1} w_h(l_1)\phi(l_1)\phi^T(l_1). \tag{1.57}$$

The formulas (1.52)-(1.57) are valid for any polynomials $\phi_k(l_1)$ even they are given in a form different from (1.7) and for any window function $w_h(l_1)$.

Here we are going to consider some particular cases of the window function w in order to demonstrate what kind of kernels we obtain for these estimates. We begin from the $1D$ case as a more simple one.

$1D$ KERNELS

Let us assume that the discrete variable polynomials $\phi(l_1)$ are orthogonal with the window $w_h(l_1)$. It means that

$$\Phi_h = I_{M\times M}.$$

In the scalar form this orthonormality condition gives the equations

$$\sum_{l_1} w_h(l_1)\phi_r(l_1)\phi_s(l_1) = \delta_{r,s},\ r,s = 1,...,\ M.$$

Then, the expressions (1.55) and (1.56) are simplified to

$$g_h(l_1) = w_h(l_1)\phi^T(0)\phi(l_1), \tag{1.58}$$

$$g_h^{(k_1)}(l_1) = (-1)^{k_1} w_h(l_1)[D^{(k_1)}\phi^T(0)]\phi(l_1), \tag{1.59}$$

and analytical formulas can be given for the kernels $g_h(l_1)$ and $g_h^{(k_1)}(l_1)$.

Let us present some special cases.

(I) **Rectangular symmetric window**

$$w_h(l_1) = \frac{1}{2h+1},\ |l_1| \le h, \tag{1.60}$$

where the parameter h is nonnegative integer, $h \ge 0$.

The discrete Lagrange polynomials of the power $m \le 2$

$$\phi_1(l_1) = 1,\ \phi_2(l_1) = \sqrt{\frac{3}{h(h+1)}}l_1, \tag{1.61}$$

$$\phi_3(l_1) = \sqrt{\frac{5}{h(h+1)(2h+3)(2h-1)}} \times$$

$$\times(3l_1^2 - h(h+1)).$$

are orthogonal with the window (1.60)

The elementary calculations give the formulas of the kernels for the smoothing and differentiation operators (1.58) and (1.59):

(a) For $m = 0$, $h \ge 0$,

$$g_h(l_1) = \frac{1}{2h+1},\ |l_1| \le h, \tag{1.62}$$

(b) For $m = 1$, $h \ge 1$,

$$g_h(l_1) = \frac{1}{2h+1},\ |l_1| \le h, \tag{1.63}$$

$$g_h^{(1)}(l_1) = \frac{-3l_1}{h(h+1)(2h+1)},\ |l_1| \le h,$$

(c) For $m = 2$, $h \geq 1$,

$$\begin{aligned} g_h(l_1) &= (1 + 5\frac{h(h+1) - 3l_1^2}{(2h+3)(2h-1)})\frac{1}{2h+1}, \ |l_1| \leq h, \\ g_h^{(1)}(l_1) &= \frac{-3l_1}{h(h+1)(2h+1)}, \ |l_1| \leq h, \\ g_h^{(2)}(l_1) &= \frac{30(3l_1^2 - h(h+1))}{h(h+1)(2h+3)(2h-1)(2h+1)}, \ |l_1| \leq h. \end{aligned} \tag{1.64}$$

It can be verified that the moment equation (1.30) for these kernels are satisfied in the form: for $g_h(l_1)$

$$\sum_{l_1=-h}^{h} g_h(l_1) = 1, \tag{1.65}$$

for $g_h^{(1)}(l_1)$

$$\begin{aligned} &\sum_{l_1=-h}^{h} g_h^{(1)}(l_1) = 0, \\ &\sum_{l_1=-h}^{h} g_h^{[1]}(l_1) \cdot l_1 = -1; \end{aligned} \tag{1.66}$$

for $g_h^{(2)}(l_1)$

$$\begin{aligned} &\sum_{l_1=-h}^{h} g_h^{(2)}(l_1) = 0, \\ &\sum_{l_1=-h}^{h} g_h^{(2)}(l_1) \cdot l_1 = 0, \\ &\sum_{l_1=-h}^{h} g_h^{(2)}(l_1) \cdot l_1^2/2 = 1. \end{aligned} \tag{1.67}$$

It is interesting to note the fact that the kernels $g_h(l_1)$ are the same for $m = 0$ and $m = 1$ (formulas (1.62) and (1.63)) and the kernels $g_h^{(1)}(l_1)$ for the first derivative estimation are the same for $m = 1$ and $m = 2$ (formulas (1.63) and (1.64)).

It can be shown that this property follows from the symmetry of the window $w_h(l_1)$ (1.60). In general, we have the same smoothing kernels $g_h(l_1)$ for $m = 2r$ and $m = 2r + 1$, $r = 0, 1, ...$, and the same differentiation kernels

$g_h^{(k)}(l_1)$ for $m = k + 2r$ and $m = k + 2r + 1$.

(II) **Rectangular nonsymmetric window**

$$w_h(l_1) = \frac{1}{h+1},\ 0 \le l_1 \le h, \tag{1.68}$$

where the parameter h is nonnegative integer, $h \ge 0$.

The discrete Lagrange polynomials of the power $m \le 2$

$$\begin{aligned} \phi_1(l_1) &= 1,\ \phi_2(l_1) = \sqrt{\frac{3}{h(h+1)}}(2l_1 - h), \qquad (1.69) \\ \phi_3(l_1) &= \sqrt{\frac{5}{h(h+2)(h+3)(h-1)}} \cdot (6l_1^2 - 6hl_1 + h(h-1)). \end{aligned}$$

are orthonormal with the window (1.68)

The elementary calculations according to (1.58) and (1.59) give the formulas for the smoothing and differentiation kernels:

(a) For $m = 0$, $h \ge 0$,

$$g_h(l_1) = \frac{1}{h+1},\ 0 \le l_1 \le h, \tag{1.70}$$

(b) For $m = 1$, $h \ge 1$,

$$\begin{aligned} g_h(l_1) &= \frac{2}{(h+2)(h+1)}[2h + 1 - 3l_1],\ 0 \le l_1 \le h, \qquad (1.71) \\ g_h^{(1)}(l_1) &= \frac{-6(2l_1 - h)}{h(h+2)(h+1)},\ 0 \le l_1 \le h, \end{aligned}$$

(c) For $m = 2$,

$$\begin{aligned} g_h(l_1) &= \frac{3(3h^2 + 3h + 2 - 6(2h+1)l_1 + 10l_1^2)}{(h+2)(h+3)(h+1)},\ 0 \le l_1 \le h, \qquad (1.72) \\ g_h^{(1)}(l_1) &= \frac{18h(h-1)(2h+1) - 6[(2h+1)(8h-3)l_1 + 30hl_1^2]}{h(h+1)^2(h+3)(h-1)},\ 0 \le l_1 \le h, \\ g_h^{(2)}(l_1) &= \frac{60(6l_1^2 - 6hl_1 + h(h-1))}{h(h+2)(h+3)(h^2-1)},\ 0 \le l_1 \le h. \end{aligned}$$

The moment conditions (1.30) can be verified for the derived kernels (1.70)-(1.72).

(III) **Exponential nonsymmetric window function**

$$w(l_1) = (1-\gamma)\gamma^{l_1},\ 0 \le l_1 < \infty,\ 0 < \gamma < 1.$$

The corresponding orthogonal polynomials

$$\begin{aligned} \phi_1(l_1) &= 1,\ \phi_2(l_1) = \frac{1}{\sqrt{\gamma}}[l_1(1-\gamma)-\gamma], \\ \phi_3(l_1) &= \frac{1}{2\gamma}[2\gamma^2-(1+3\gamma)(1-\gamma)l_1+(1+\gamma)^2l_1^2]. \end{aligned} \tag{1.73}$$

Calculations give the formulas for the kernels of the smoothing operators (1.58):

(a) For $m = 0$, $h \geq 0$,

$$g_h(l_1) = w(l_1),\ 0 \leq l_1 < \infty,$$

(b) For $m = 1$, $h \geq 1$,

$$g_h(l_1) = [1+\gamma-(1-\gamma)l_1]w(l_1),\ 0 \leq l_1 < \infty,$$

(c) For $m = 2$,

$$g_h(l_1) = \frac{1}{2}[2(1+\gamma+\gamma^2)-3(1-\gamma^2)l_1+(1-\gamma)^2l_1^2]w(l_1),\ 0 \leq l_1 < \infty.$$

We do not show here bulky kernels of the differentiation operators, which also can be derived according to the described procedure.

$2D$ KERNELS

Let the $2D$ window w_h be a multiplicative function of integers l_1 and l_2:

$$w_h(l_1, l_2) = w_{h,1}(l_1) w_{h,2}(l_2), \tag{1.75}$$

where $w_{h,1}(l_1)$ and $w_{h,2}(l_2)$ are $1D$ functions of integers l_1 and l_2.

Assume that $\phi_{j,1}(l_1)$ and $\phi_{j,2}(l_2)$, $j = 1, 2, ..., m+1$ be $1D$ polynomials of the power $j-1$ orthogonal with the weights $w_{h,1}(l_1)$ and $w_{h,2}(l_2)$ respectively.

Then, the $2D$ polynomials orthogonal with the window $w_h(l_1, l_2)$ (1.75) can be derived from these $1D$ polynomials. A set of these $2D$ polynomial $\phi(l_1, l_2)$ of the power $k-1$ can be composed from $1D$ polynomials as a direct product of the sets of the $1D$ orthogonal polynomials

$$\begin{aligned} &\phi^{(j_1,j_2)}(l_1, l_2) = \phi_{j_1,1}(l_1)\phi_{j_2,2}(l_2), \\ &j_1 = 1, ..., k-1,\ j_2 = 1, ..., k-1,\ j_1 + j_2 = k-1, \end{aligned} \tag{1.76}$$

where positive integers $j_1 \geq 1$ and $j_2 \geq 1$ take all values satisfying the equation $j_1 + j_2 = k - 1$.

These $2D$ polynomials built successively for $k - 1 = 0, 1, ..., m$ form a complete set of $2D$ polynomials of the power from 0 through m.

It can be verified that these $2D$ polynomials are orthogonal with the $2D$ window $w_h(l_1, l_2)$ (1.75).

These $2D$ orthogonal polynomials can be ordered according to their powers similarly to (1.7)

$$\begin{aligned} &\phi_1(l_1, l_2) = \phi_{1,1}(l_1)\phi_{1,2}(l_2), \text{ for } m = 0; \\ &\phi_2(l_1, l_2) = \phi_{2,1}(l_1)\phi_{1,2}(l_2),\ \phi_3(l_1, l_2) = \phi_{1,1}(l_1)\phi_{2,2}(l_2), \text{ for } m = 1; \\ &\ \phi_4(l_1, l_2) = \phi_{3,1}(l_1)\phi_{1,2}(l_2),\ \phi_5 = \phi_{1,1}(l_1)\phi_{3,2}(l_2), \\ &\ \phi_6(l_1, l_2) = \phi_{2,1}(l_1)\phi_{2,2}(l_2), \text{ for } m = 2; \\ &..., \end{aligned} \tag{1.77}$$

with the total number of the polynomials equal to $M = \frac{(2+m)!}{2 \cdot m!}$.

In this way we design a complete set of the $2D$ polynomials of the power less or equal to m and orthogonal with the window $w_h(l_1, l_2)$. The estimator kernels can be obtained from the formulas (1.52) and (1.53), which are general for the $2D$ case and where Φ_h is the identity matrix and omitted. All calculations are similar to used above for the $1D$ case.

Let us present a few $2D$ kernels derived in this way using the $1D$ orthogonal polynomials applied above for the derivation of the kernels of the $1D$ estimates. It is assumed in what follows that

$$w_{h_1,h_2}(l_1, l_2) = w_{h_1}(l_1) w_{h_2}(l_2)$$

with possibly different h_1 and h_2 for the variables l_1 and l_2.

(I) **Rectangular symmetric windows with**

$$w_{h_j}(l_j) = \frac{1}{2h_j + 1}, \ |l_j| \leq h_j, \, j = 1, 2.$$

and

$$w_{h_1,h_2}(l_1, l_2) = w_{h_1}(l_1) w_{h_2}(l_2) = \frac{1}{(2h_1 + 1)(2h_2 + 1)}, \ |l_j| \leq h_j, \ j = 1, 2,$$

Using the orthogonal polynomials (1.61) and the kernel formulas (1.52), (1.53) we derive the kernels of the operators of function and first derivative estimation for the different powers m.

(a) For $m = 0$, $h_j \geq 0$,

$$g_{h_1,h_2}(l_1, l_2) = w_{h_1,h_2}(l_1, l_2), \ 0 \leq l_j \leq h_j, \ j = 1, 2, \tag{1.78}$$

(b) For $m = 1$, $h_j \geq 1$,

$$\begin{aligned} g_{h_1,h_2}(l_1, l_2) &= w_{h_1,h_2}(l_1, l_2), \ 0 \leq l_j \leq h_j, && (1.79) \\ g^{(1,0)}_{h_1,h_2}(l_1, l_2) &= \frac{-3l_1}{h_1(h_1 + 1)} w_{h_1,h_2}(l_1, l_2), \ 0 \leq l_j \leq h_j, && (1.80) \\ g^{(0,1)}_{h_1,h_2}(l_1, l_2) &= \frac{-3l_2}{h_2(h_2 + 1)} w_{h_1,h_2}(l_1, l_2), \ 0 \leq l_j \leq h_j, && (1.81) \end{aligned}$$

(c) For $m = 2$, $h_j \geq 1$,

$$g_{h_1,h_2}(l_1, l_2) = (1 + 5 \sum_{j=1}^{2} \frac{h_j(h_j + 1) - 3l_j^2}{(2h_j + 3)(2h_j - 1)}) \cdot w_{h_1,h_2}(l_1, l_2), \tag{1.82}$$

$0 \leq l_j \leq h_j$,

$$g^{(1,0)}_{h_1,h_2}(l_1, l_2) = \frac{-3l_1}{h_1(h_1 + 1)} w_{h_1,h_2}(l_1, l_2), \ 0 \leq l_j \leq h_j, \tag{1.83}$$

$$g^{(0,1)}_{h_1,h_2}(l_1, l_2) = \frac{-3l_2}{h_2(h_2 + 1)} w_{h_1,h_2}(l_1, l_2), \ 0 \leq l_j \leq h_j. \tag{1.84}$$

It deserves to be noted that the smoothing operators (1.78) and (1.79) are identical to each other. A similar coincidence of (1.80) with (1.83) as well as (1.81) with (1.84) have a place for first derivative estimation.

This coincidence of the kernels corresponding to different powers of the polynomial approximation is a result of the window symmetry property, $w_{h_1,h_2}(l_1, l_2) = w_{h_1,h_2}(-l_1, -l_2)$, discussed above for the $1D$ case.

(II) **Rectangular nonsymmetric windows with**

$$w_{h_j}(l_j) = \frac{1}{h_j + 1},\ 0 \leq l_j \leq h_j,\ j = 1, 2.$$

and

$$w_{h_1,h_2}(l_1, l_2) = w_{h_1}(l_1)w_{h_2}(l_2) = \frac{1}{h_1 + 1}\frac{1}{h_2 + 1},\ 0 \leq l_j \leq h_j,\ j = 1, 2,$$

Using the orthogonal polynomials (1.69) and the kernel formulas (1.52), (1.53) we derive the kernels of the operators of function and first derivative estimation for powers $m = 0, 1$:.

(a) For $m = 0$, $h_j \geq 0$,

$$g_{h_1,h_2}(l_1, l_2) = w_{h_1,h_2}(l_1, l_2),\ 0 \leq l_j \leq h_j,\ j = 1, 2,$$

(b) For $m = 1$, $h_j \geq 1$,

$$g_{h_1,h_2}(l_1, l_2) = (1 + 3\sum_{s=1}^{2} \frac{h_j - 2l_j}{h_j + 2})w_{h_1,h_2}(l_1, l_2),\ 0 \leq l_j \leq h_j,$$
$$j = 1, 2,$$
$$g^{(1,0)}_{h_1,h_2}(l_1, l_2) = -\frac{6(2l_1 - h_1)}{h_1(h_1 + 2)}w_{h_1,h_2}(l_1, l_2),\ 0 \leq l_j \leq h_j,\ j = 1, 2,$$
$$g^{(0,1)}_{h_1,h_2}(l_1, l_2) = -\frac{6(2l_2 - h_2)}{h_2(h_2 + 2)}w_{h_1,h_2}(l_1, l_2),\ 0 \leq l_j \leq h_j,\ j = 1, 2,$$

Many others multivariable kernel functions can be given in the analytical form even for higher values of the power m (e.g. [59]). However, we are not going to present these result here as the formulas become more complex for $m > 1$. Actually, these analytical representations of the kernel are not very useful as all necessary calculation can be produced numerically based on the general formulas (1.40) and (1.42). Even the orthogonality of the polynomials is not very helpful as finally the kernels can be given as numerical matrixes of their values.

1.4.2 Numerical regular grid kernels

Here we present some examples of the homogeneous kernels for smoothing and differentiation.

The formulas (1.40) and (1.42) with $\Delta_1 = \Delta_2 = 1$ are applied for calculations. The results are given in two different forms. One is graphical illustrative used for visualization of a distribution of numerical values of the weights in the kernel (mask). In parallel, we present the mask weights also in tables in order to show the weights in detail.

We consider finite masks only. The masks can be symmetric or non-symmetric with respect to the center of the LPA denoted by x. The symmetric mask has x as a center point. The non-symmetric masks are shown in Figure 1.1 as quadrants with the corresponding vertex at the point x. All mask are assumed to be square of the size $h \times h$. We consider two types of the window w_h constant and Gaussian truncated to the size of the mask.

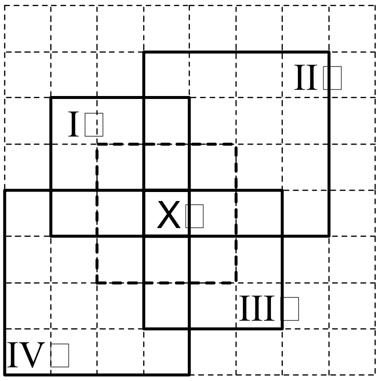


Figure 1.1: Graphical interpretation of the symmetric and non-symmetric quadrant masks: $'x'$ corresponds to the estimation pixel, I, II, III, IV show the quadrant masks ($Q = 1, 2, 3, 4$). The symmetric mask ($Q = 0$) is centered with respect to $'x'$.

The visualization of the symmetric masks is given in Figures 1.2 and 1.3, as well as in Figures 1.6 and 1.7, while the corresponding numerical weights are shown in Figures 1.10 and 1.11, and in Figures 1.15 and 1.16, respectively. Details of the masks and the parameters of the used LPA models are specified in the comments to the figures.

The power $m = 0$ results in a trivial estimator since it leads to the standard sample mean of the data in the window. We consider the powers $m \geq 1$ in order to obtain non-trivial weights and to have an ability to estimate the derivatives.

In Figure 1.2 the smoothing kernel given for $m = 1$ is constant inside the mask. The same kernel could be obtained by using the zero power LPA with $m = 0$. The masks for the derivative estimation on X and on Y (marked as "DER ON X" and DER ON Y") have a clear linear structure with the vertical and horizontal bands corresponding equal values of the weights. It shows that the LPA gives a smoothed finite-difference estimate of the derivatives. We wish to note that in this section we use X and Y in order to mark a direction of differentiation.

The masks in Figure 1.3 are obtained for $m = 2$. They demonstrate a more complex structure of the kernel values. The quadratic weights with a peak at the center of the mask are easily recognized in the first imaging of the weights given for the smoothing estimator. It is interesting to note once more that the weights for estimation of the derivatives $\partial/\partial X$ and $\partial/\partial Y$ are exactly the same as those for these derivatives in Figure 1.2 where the power $m=1$ is used. Again this coincidence of the estimates of the different orders follows from the symmetry of the used window function.

Masks shown in Figures 1.6 and 1.7 are different from those given in Figures 1.2 and 1.3 by using the Gaussian window function instead of the constant one. The Gaussian window function results in a larger variety of the kernel values used in the mask. However, the symmetry of the weights as well as the coincidence of the estimates obtained for $m=1$ and $m=2$ discussed above can be easily noticed.

A clear linear band structure of the weights for the derivatives $\partial/\partial X$, $\partial/\partial Y$, $\partial^2/\partial X^2$, $\partial^2/\partial Y^2$ in the mask with the constant window function corresponds to the conventional idea of using weighted finite differences for derivative estimation. The LPA with the symmetric window function formalizes this idea. For the Gaussian windows the structure of the weights becomes non-trivial and more complex. The numerical weights of the masks shown in Figures 1.6 and 1.7 can be seen in Figures 1.15 and 1.16.

The nonsymmetric masks $h \times h$ are shown for Quadrant 1 (Q1 mask) in Figures 1.4, 1.5, and in Figures 1.8, 1.9 as blocks of the square larger size masks $(2h-1)\times(2h-1)$. This larger size square can be treated as an extended mask of the estimator. The LPA reference point x is at the center of this extended mask. Zero values are assumed for the weights of the extended mask beyond the considered Quadrant 1. These zero values are visualized in a grey color.

Actually, there are two motivations behind using these larger size masks. The first one is an intention to emphasize the non-symmetric nature of the quadrant estimators. The second one is more pragmatic and important concerning the implementation of the estimator. The MATLAB program conv2.m with the symmetric central location of the reference point x of the mask fits perfectly to the correct implementation of the LPA estimators using the convolution operation.

The numerical weights of the masks shown in Figures 1.4, 1.5, 1.8, 1.9 can be seen in Figures 1.12, 1.13, 1.14, 1.17 and 1.18, 1.19, respectively.

The most of the classical differentiation and smoothing masks can be treated as the LPA derived masks for some LPA parameters. Let us consider well know examples.

(a) The Roberts (2×2) differentiation mask

$$P_{Roberts}=\left[\begin{array}{cc} 0 & 1 \\ -1 & 0 \end{array}\right] \tag{1.85}$$

can be treated as a non-symmetric mask with the kernel matrix

$$P_{LPA} = g_h^{[1,0]} + g_h^{[0,1]},$$

where $g_h^{[1,0]}$ and $g_h^{[0,1]}$ are derived for $m = 1$ and $h = 2$.

Then, the matrices $g_h^{[1,0]}$, $g_h^{[0,1]}$ and P_{LPA} are as follows

$$\begin{aligned} P_{LPA} &= g_2^{[1,0]} + g_2^{[0,1]} = \begin{pmatrix} -0.5 & 0.5 & 0 \\ -0.5 & 0.5 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0.5 & 0.5 & 0 \\ -0.5 & -0.5 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ P_{LPA} &= \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \end{aligned} \tag{1.86}$$

The 2×2 sub-matrix in the left upper corner of the matrix P_{LPA} is equal to the matrix $P_{Roberts}$. Thus, we install a link between the $Roberts'$ mask and the LPA differentiation masks. It shows that the $P_{Roberts}$ mask gives the estimate of the sum of the derivatives on X and on Y, i.e. the estimate of $\partial y(x)/\partial X + \partial y(x)/\partial Y$.

Note, that the $Roberts'$ mask, being of the size (2×2), defines the estimate as a finite difference of the image intensity however the reference point of this estimate is not specified. Here we have a clear advantage of the LPA with a non-symmetric mask which actually makes a difference between reference point and ordinary point of the mask and enables one to obtain a clear interpretation of any sort of finite difference schemes.

(b) The Prewitt's mask

$$P_{\text{Prewitt}} = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

within a scalar factor coincides with the LPA mask obtained for the symmetric window (3×3) and $m = 1$

$$g_3^{[0,1]} = \begin{pmatrix} 0.1667 & 0.1667 & 0.1667 \\ 0 & 0 & 0 \\ -0.1667 & -0.1667 & -0.1667 \end{pmatrix}.$$

Thus,

$$P_{\text{Prewitt}} = -6g_h^{[0,1]}.$$

(c) The Sobel's differentiation mask

$$P_{\text{Sobel}} = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

can be presented as the sum

$$P_{\text{Sobel}} = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix},$$

where the mask $\begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ corresponds to $1D$ differentiation on Y. The last mask can be derived by use of $1D$ LPA differentiator. In this way the Sobel's differentiation mask can be presented in the form of the linear combination of two LPA masks: one for $2D$ differentiation $g_3^{[0,1]}$, and another one for $1D$ differentiation.

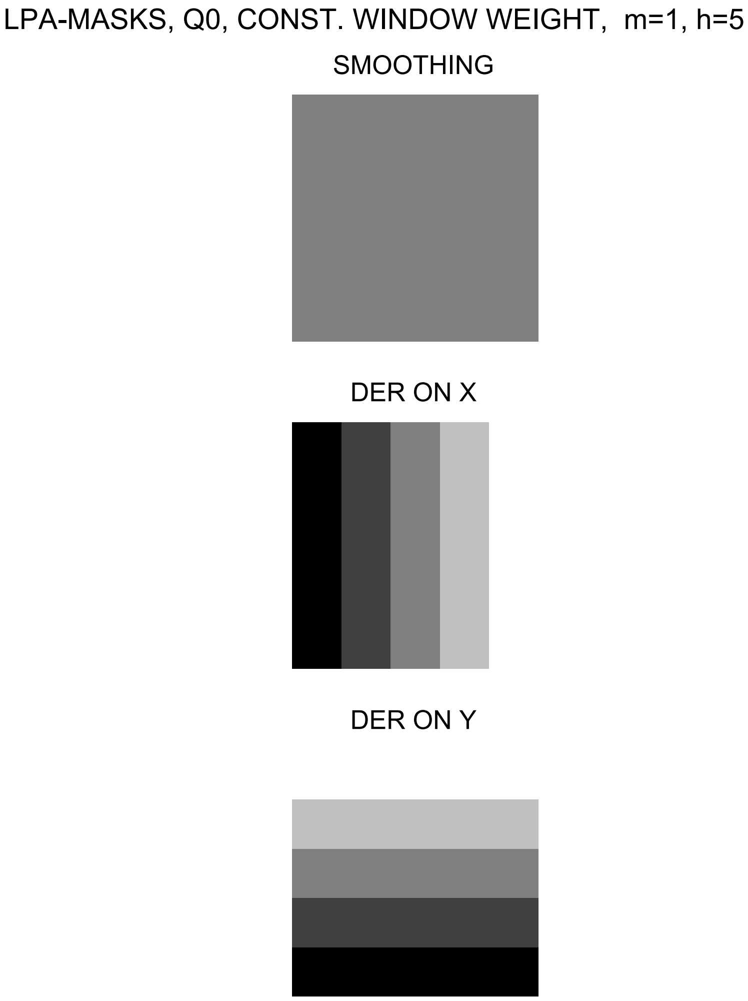


Figure 1.2: Symmetric LPA masks for estimation (smoothing) and differentiation. The LPA window weight w is constant, the power $m = 1$. The mask size is 5×5, $h = 5$. The reference pixel is at the center of the square. Black and white define a scale between smaller and larger values of the kernel values.

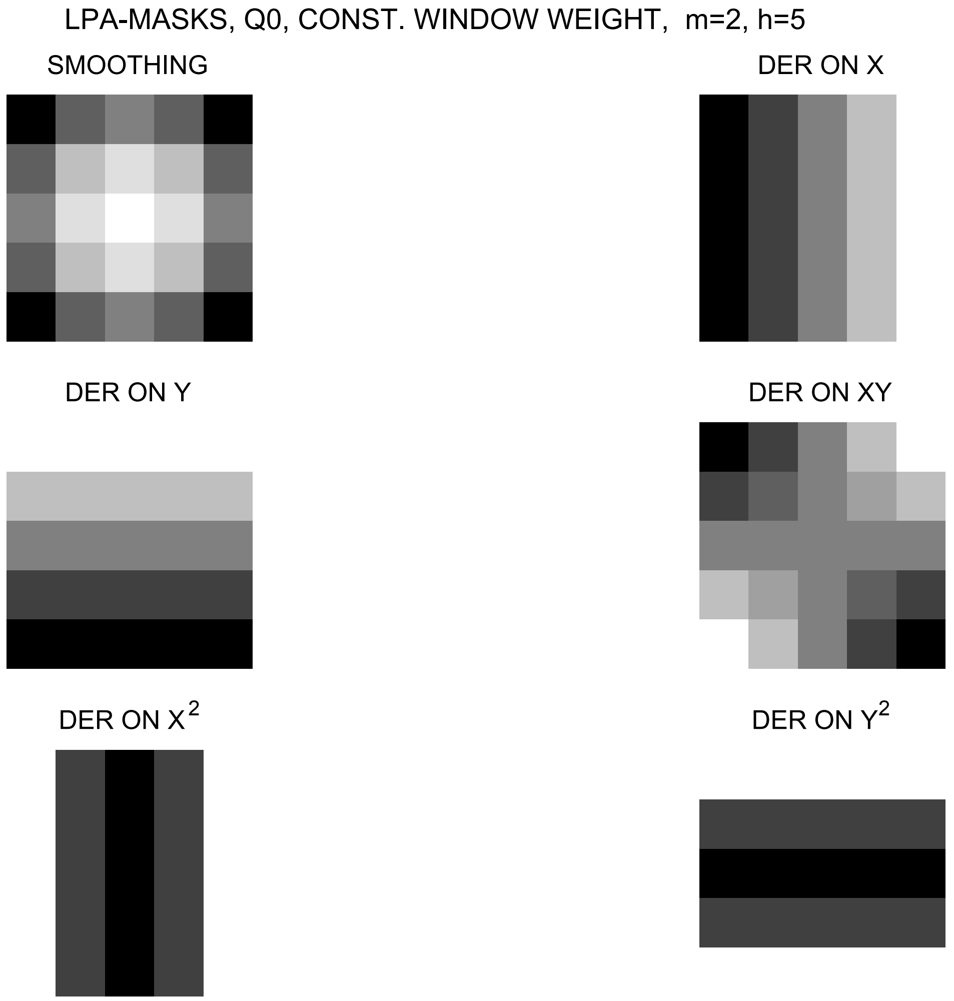


Figure 1.3: Symmetric LPA masks for estimation (smoothing) and differentiation. The LPA window weight w is constant, the power $m = 2$. The mask size is 5×5, $h = 5$. The reference pixel is at the center of the mask. Black and white define a scale between smaller and larger values of the kernel values.

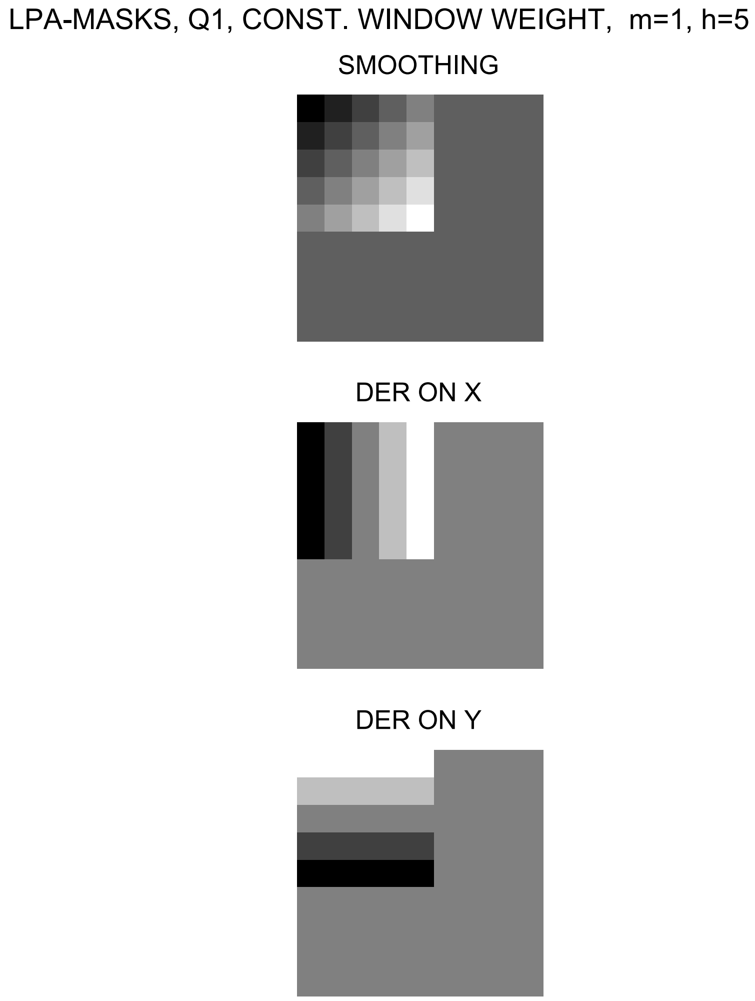
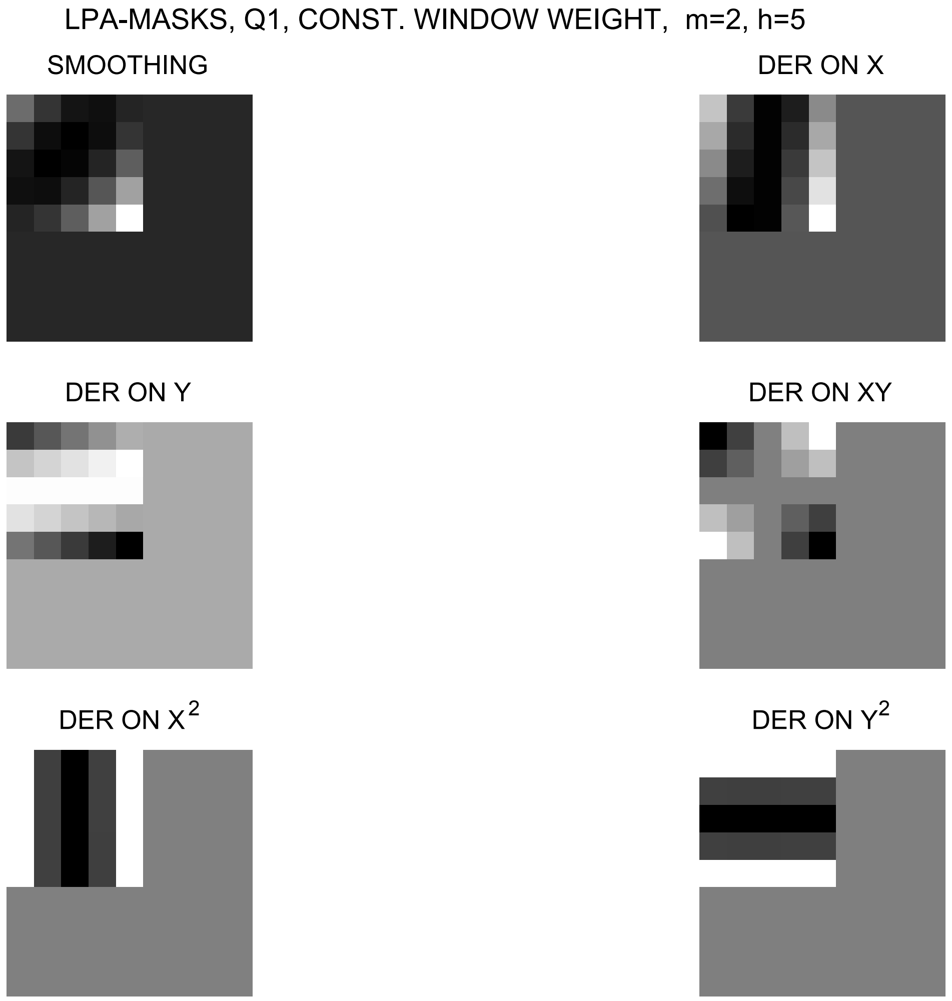


Figure 1.4: Non-symmetric LPA masks for estimation (smoothing) and differentiation. The LPA window weight w is constant, the first quadrant. The reference LPA pixel is at the right down corner of the estimation mask depicted into the extended mask (larger size grey square 11×11). This grey color corresponds to zero values of the weights of the extended mask. The LPA power $m = 1$. The estimation mask size is 5×5, $h = 5$. Black and white define a scale between smaller and larger values of the kernel values.



LPA-MASKS, Q1, CONST. WINDOW WEIGHT, m=2, h=5

SMOOTHING

DER ON X

DER ON Y

DER ON XY

DER ON X^2

DER ON Y^2

Figure 1.5: Non-symmetric LPA masks for estimation (smoothing) and differentiation. The LPA window weight w is constant, the first quadrant, the power $m = 2$. The reference (estimation) pixel is at the right down corner of mask depicted into the larger extended mask (grey square 11×11). This grey color corresponds to zero values of the mask values. The estimation mask size is 5×5, $h = 5$. Black and white define a scale between smaller and larger values of the kernel weight values respectively.

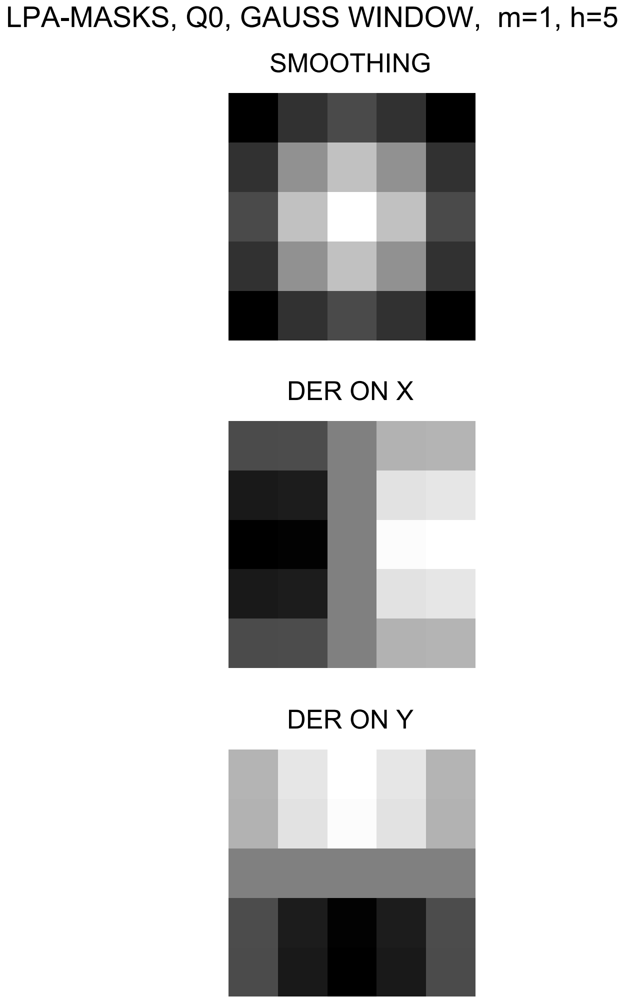


Figure 1.6: Symmetric LPA masks for estimation (smoothing) and differentiation. The LPA window weight w is Gaussian, the power $m = 1$. The mask size is 5×5, $h = 5$. The reference pixel is at the center of the mask. Black and white define a scale between smaller and larger values of the kernel values.

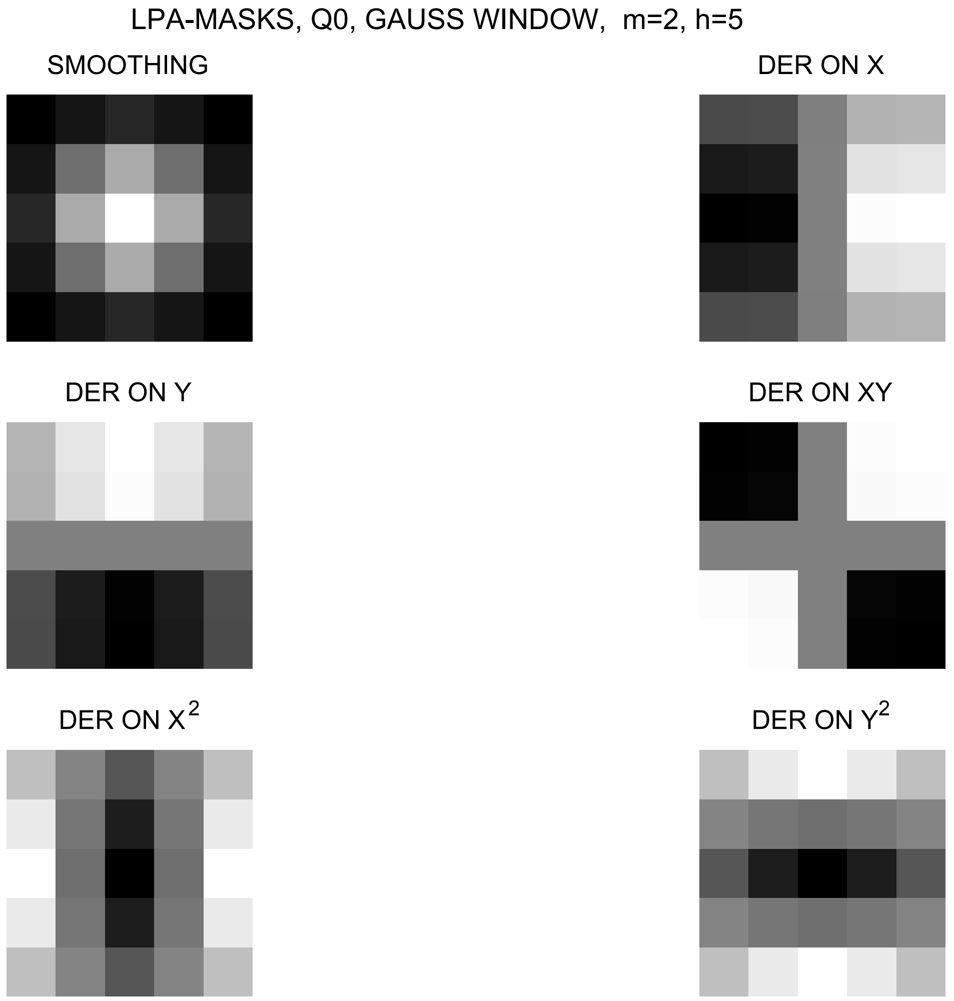


Figure 1.7: Symmetric LPA masks for estimation (smoothing) and differentiation. The LPA window weight w is Gaussian, the power $m = 2$. The mask size is 5×5, $h = 5$. The reference pixel is at the center of the mask. Black and white define a scale between smaller and larger values of the kernel values.

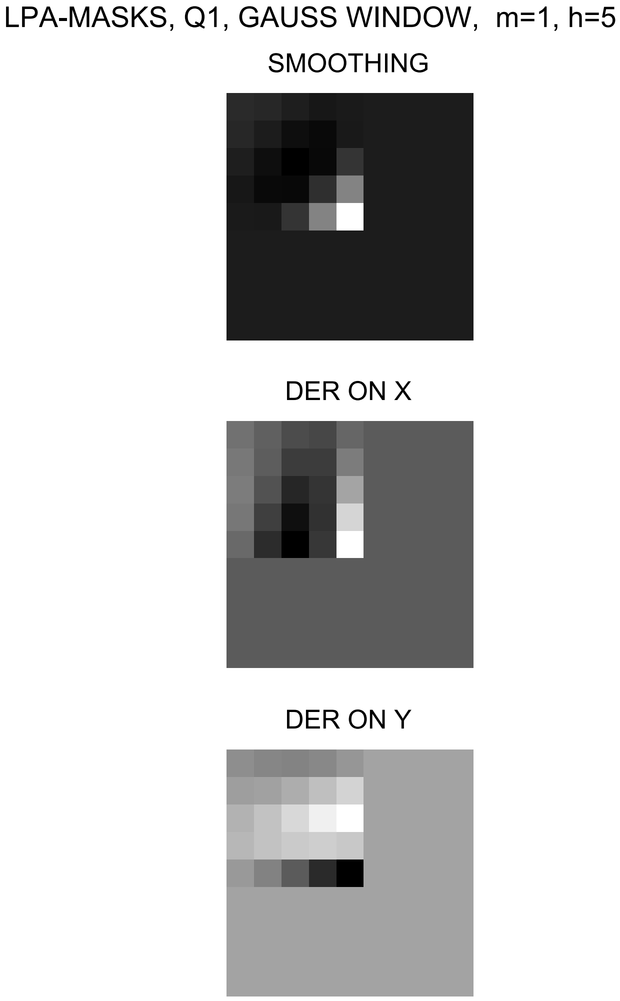


Figure 1.8: Non-symmetric LPA masks for estimation (smoothing) and differentiation. The LPA window weight w is Gaussian, the first quadrant. The reference pixel is at the right down corner of mask depicted into the larger extended mask (grey square 11×11). The grey color corresponds to zero values of the mask values. The LPA power $m = 1$. The estimation mask size is 5×5, $h = 5$. Black and white define a scale between smaller and larger values of the kernel values.

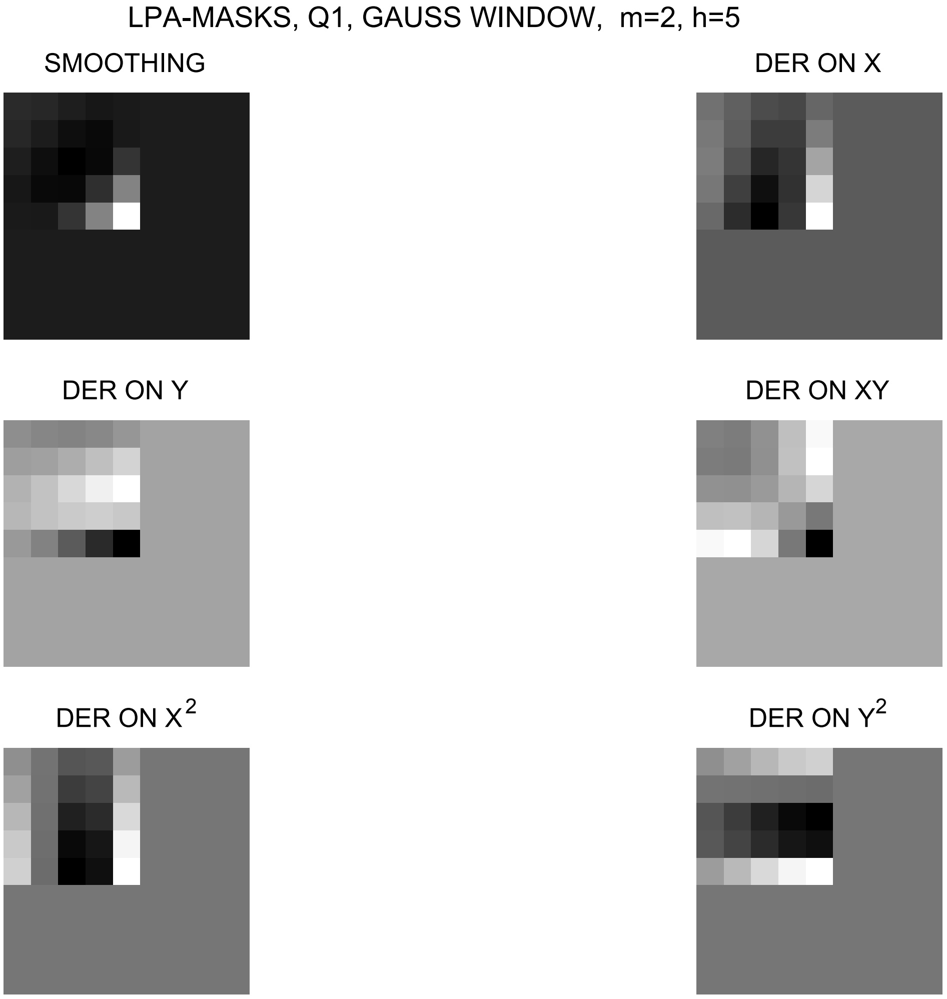
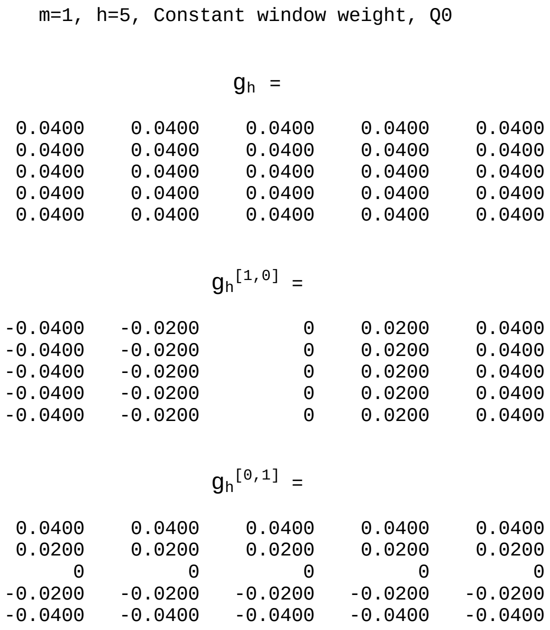
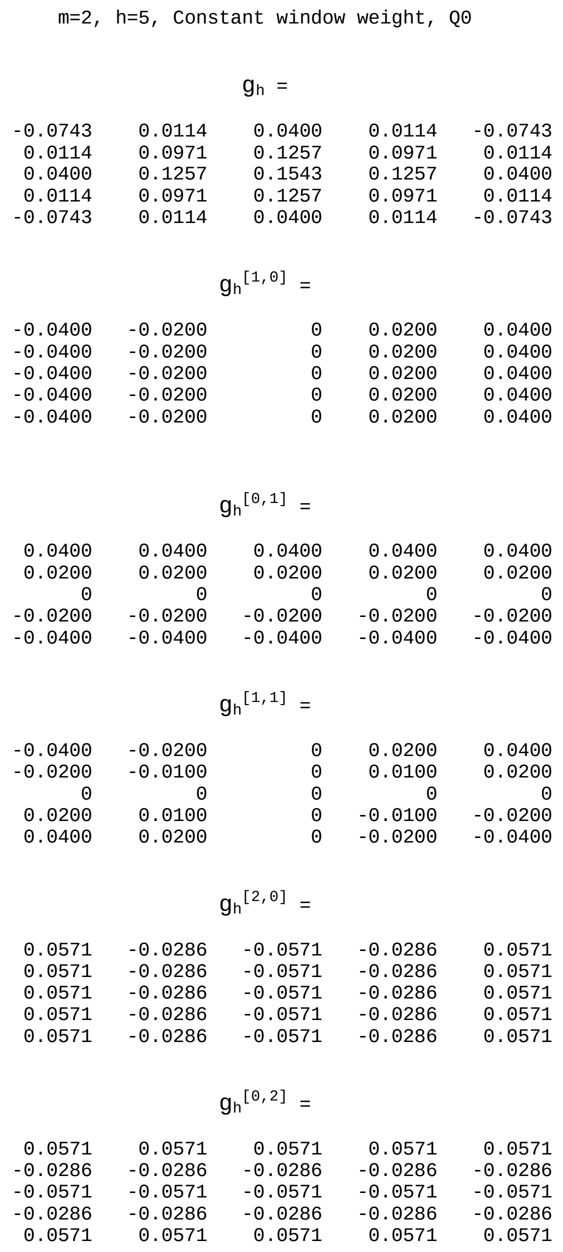


Figure 1.9: Non-symmetric LPA masks for estimation (smoothing) and differentiation. The LPA window weight w is Gaussian, the first quadrant. The reference pixel is at the right down corner of the mask depicted into the larger grey extended mask (square 11×11). This grey color corresponds to zero values of the mask weights. The LPA power $m = 1$. The estimation mask size is 5×5, $h = 5$. Black and white define a scale between smaller and larger values of the kernel values.



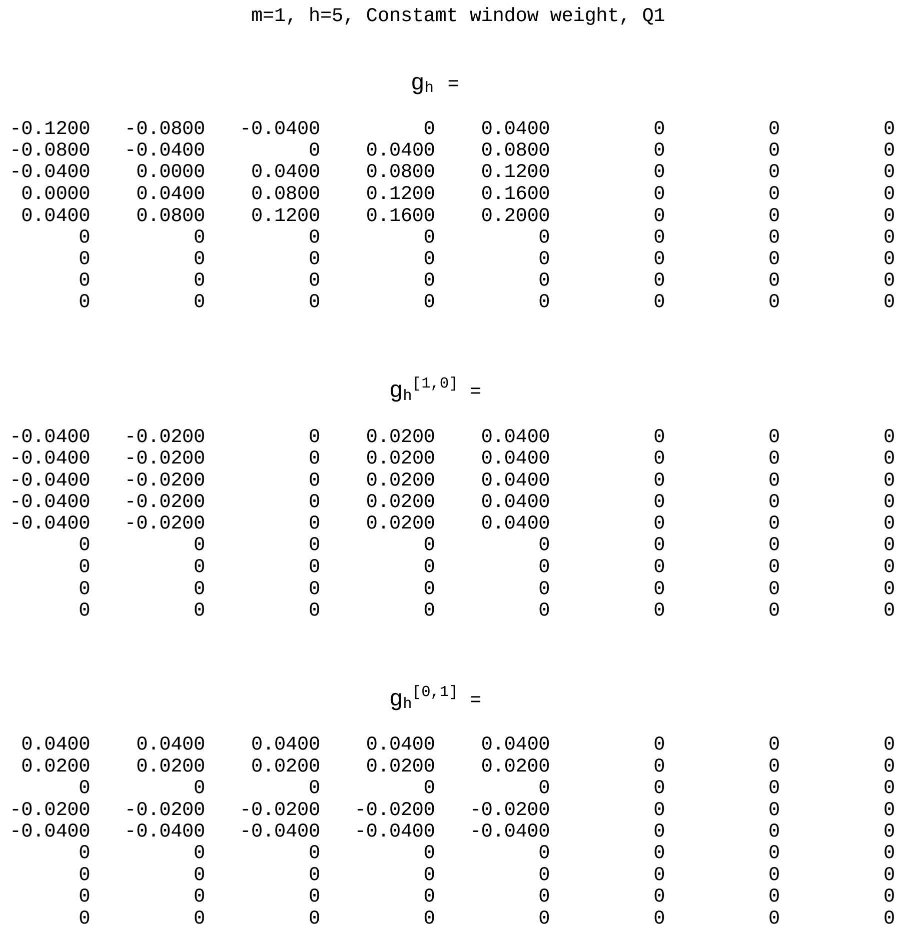
```
        m=1, h=5, Constant window weight, Q0

                      g_h =

   0.0400    0.0400    0.0400    0.0400    0.0400
   0.0400    0.0400    0.0400    0.0400    0.0400
   0.0400    0.0400    0.0400    0.0400    0.0400
   0.0400    0.0400    0.0400    0.0400    0.0400
   0.0400    0.0400    0.0400    0.0400    0.0400

                    g_h^[1,0] =

  -0.0400   -0.0200         0    0.0200    0.0400
  -0.0400   -0.0200         0    0.0200    0.0400
  -0.0400   -0.0200         0    0.0200    0.0400
  -0.0400   -0.0200         0    0.0200    0.0400
  -0.0400   -0.0200         0    0.0200    0.0400

                    g_h^[0,1] =

   0.0400    0.0400    0.0400    0.0400    0.0400
   0.0200    0.0200    0.0200    0.0200    0.0200
        0         0         0         0         0
  -0.0200   -0.0200   -0.0200   -0.0200   -0.0200
  -0.0400   -0.0400   -0.0400   -0.0400   -0.0400
```

Figure 1.10: Symmetric LPA masks for estimation (smoothing) and differentiation. The LPA window weight is constant, the power $m = 1$. The mask size is 5×5, $h = 5$. The reference pixel is at the center of the square.



```
              m=2, h=5, Constant window weight, Q0

                         gh =

  -0.0743    0.0114    0.0400    0.0114   -0.0743
   0.0114    0.0971    0.1257    0.0971    0.0114
   0.0400    0.1257    0.1543    0.1257    0.0400
   0.0114    0.0971    0.1257    0.0971    0.0114
  -0.0743    0.0114    0.0400    0.0114   -0.0743


                       gh[1,0] =

  -0.0400   -0.0200         0    0.0200    0.0400
  -0.0400   -0.0200         0    0.0200    0.0400
  -0.0400   -0.0200         0    0.0200    0.0400
  -0.0400   -0.0200         0    0.0200    0.0400
  -0.0400   -0.0200         0    0.0200    0.0400



                       gh[0,1] =

   0.0400    0.0400    0.0400    0.0400    0.0400
   0.0200    0.0200    0.0200    0.0200    0.0200
        0         0         0         0         0
  -0.0200   -0.0200   -0.0200   -0.0200   -0.0200
  -0.0400   -0.0400   -0.0400   -0.0400   -0.0400


                       gh[1,1] =

  -0.0400   -0.0200         0    0.0200    0.0400
  -0.0200   -0.0100         0    0.0100    0.0200
        0         0         0         0         0
   0.0200    0.0100         0   -0.0100   -0.0200
   0.0400    0.0200         0   -0.0200   -0.0400


                       gh[2,0] =

   0.0571   -0.0286   -0.0571   -0.0286    0.0571
   0.0571   -0.0286   -0.0571   -0.0286    0.0571
   0.0571   -0.0286   -0.0571   -0.0286    0.0571
   0.0571   -0.0286   -0.0571   -0.0286    0.0571
   0.0571   -0.0286   -0.0571   -0.0286    0.0571


                       gh[0,2] =

   0.0571    0.0571    0.0571    0.0571    0.0571
  -0.0286   -0.0286   -0.0286   -0.0286   -0.0286
  -0.0571   -0.0571   -0.0571   -0.0571   -0.0571
  -0.0286   -0.0286   -0.0286   -0.0286   -0.0286
   0.0571    0.0571    0.0571    0.0571    0.0571
```

Figure 1.11: Symmetric LPA masks for estimation (smoothing) and differentiation. The LPA window weight w is constant, the power $m = 2$. The mask size is 5×5, $h = 5$. The reference pixel is at the center of the mask.



```
                         m=1, h=5, Constamt window weight, Q1

                                      gh =

  -0.1200   -0.0800   -0.0400         0    0.0400         0         0         0
  -0.0800   -0.0400         0    0.0400    0.0800         0         0         0
  -0.0400    0.0000    0.0400    0.0800    0.1200         0         0         0
   0.0000    0.0400    0.0800    0.1200    0.1600         0         0         0
   0.0400    0.0800    0.1200    0.1600    0.2000         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0


                                    gh[1,0] =

  -0.0400   -0.0200         0    0.0200    0.0400         0         0         0
  -0.0400   -0.0200         0    0.0200    0.0400         0         0         0
  -0.0400   -0.0200         0    0.0200    0.0400         0         0         0
  -0.0400   -0.0200         0    0.0200    0.0400         0         0         0
  -0.0400   -0.0200         0    0.0200    0.0400         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0


                                    gh[0,1] =

   0.0400    0.0400    0.0400    0.0400    0.0400         0         0         0
   0.0200    0.0200    0.0200    0.0200    0.0200         0         0         0
        0         0         0         0         0         0         0         0
  -0.0200   -0.0200   -0.0200   -0.0200   -0.0200         0         0         0
  -0.0400   -0.0400   -0.0400   -0.0400   -0.0400         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0
```

Figure 1.12: Non-symmetric LPA masks for estimation (smoothing) and differentiation. The LPA window weight w is constant, the first quadrant. The reference pixel is at the right down corner of the non-zero estimation mask.

```
m=2, h=5, Constant window weight, Q1,
Smoothing and derivatives d/dx, d/dy
```

$g_h =$

```
 0.1543    0.0286   -0.0400   -0.0514   -0.0057         0         0         0
 0.0286   -0.0571   -0.0857   -0.0571    0.0286         0         0         0
-0.0400   -0.0857   -0.0743   -0.0057    0.1200         0         0         0
-0.0514   -0.0571   -0.0057    0.1029    0.2686         0         0         0
-0.0057    0.0286    0.1200    0.2686    0.4743         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
```

$g_h^{[1,0]}=$

```
 0.1543   -0.0371   -0.1143   -0.0771    0.0743         0         0         0
 0.1143   -0.0571   -0.1143   -0.0571    0.1143         0         0         0
 0.0743   -0.0771   -0.1143   -0.0371    0.1543         0         0         0
 0.0343   -0.0971   -0.1143   -0.0171    0.1943         0         0         0
-0.0057   -0.1171   -0.1143    0.0029    0.2343         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
```

$g_h^{[0,1]}=$

```
-0.1543   -0.1143   -0.0743   -0.0343    0.0057         0         0         0
 0.0371    0.0571    0.0771    0.0971    0.1171         0         0         0
 0.1143    0.1143    0.1143    0.1143    0.1143         0         0         0
 0.0771    0.0571    0.0371    0.0171   -0.0029         0         0         0
-0.0743   -0.1143   -0.1543   -0.1943   -0.2343         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
```

Figure 1.13: Non-symmetric *LPA* masks for estimation (smoothing) and differentiation. The *LPA* window weight w is square constant, the first quadrant. The reference (estimation) pixel is at the left down corner of the non-zero mask.

(Continuation)
m=2, h=5, Constant window weight, Q1,
Derivatives d^2/dxdy, d^2/dxdx, d^2/dydy

$g_h^{[1,1]}=$

```
-0.0400   -0.0200    0.0000    0.0200    0.0400         0         0         0
-0.0200   -0.0100   -0.0000    0.0100    0.0200         0         0         0
-0.0000   -0.0000   -0.0000   -0.0000   -0.0000         0         0         0
 0.0200    0.0100   -0.0000   -0.0100   -0.0200         0         0         0
 0.0400    0.0200    0.0000   -0.0200   -0.0400         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
```

$g_h^{[2,0]}=$

```
 0.0571   -0.0286   -0.0571   -0.0286    0.0571         0         0         0
 0.0571   -0.0286   -0.0571   -0.0286    0.0571         0         0         0
 0.0571   -0.0286   -0.0571   -0.0286    0.0571         0         0         0
 0.0571   -0.0286   -0.0571   -0.0286    0.0571         0         0         0
 0.0571   -0.0286   -0.0571   -0.0286    0.0571         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
```

$g_h^{[0,2]}=$

```
 0.0571    0.0571    0.0571    0.0571    0.0571         0         0         0
-0.0286   -0.0286   -0.0286   -0.0286   -0.0286         0         0         0
-0.0571   -0.0571   -0.0571   -0.0571   -0.0571         0         0         0
-0.0286   -0.0286   -0.0286   -0.0286   -0.0286         0         0         0
 0.0571    0.0571    0.0571    0.0571    0.0571         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
      0         0         0         0         0         0         0         0
```

Figure 1.14: Non-symmetric LPA masks for differentiation. The LPA window weight w is constant, the first quadrant, $m = 2$. The reference (estimation) pixel is at right down corner of the non-zero mask.

```
m=1, h=5, Gaussian window weight, Q0
```

g_h =

```
 0.0014    0.0090    0.0168    0.0090    0.0014
 0.0090    0.0576    0.1067    0.0576    0.0090
 0.0168    0.1067    0.1979    0.1067    0.0168
 0.0090    0.0576    0.1067    0.0576    0.0090
 0.0014    0.0090    0.0168    0.0090    0.0014
```

$g_h^{[1,0]}$ =

```
-0.0036   -0.0116   -0.0000    0.0116    0.0036
-0.0231   -0.0737   -0.0000    0.0737    0.0231
-0.0429   -0.1366   -0.0000    0.1366    0.0429
-0.0231   -0.0737   -0.0000    0.0737    0.0231
-0.0036   -0.0116   -0.0000    0.0116    0.0036
```

$g_h^{[0,1]}$ =

```
 0.0036    0.0231    0.0429    0.0231    0.0036
 0.0116    0.0737    0.1366    0.0737    0.0116
-0.0000   -0.0000   -0.0000   -0.0000   -0.0000
-0.0116   -0.0737   -0.1366   -0.0737   -0.0116
-0.0036   -0.0231   -0.0429   -0.0231   -0.0036
```

Figure 1.15: Symmetric LPA masks for estimation (smoothing) and differentiation. The LPA window weight w is Gaussian, the power $m = 1$. The mask size is 5×5, $h = 5$. The reference pixel is at the center of the mask.

```
m=2, h=5, Gaussian window weight, Q0
```

$g_h =$

```
-0.0052   -0.0135   -0.0129   -0.0135   -0.0052
-0.0135    0.0393    0.1503    0.0393   -0.0135
-0.0129    0.1503    0.4225    0.1503   -0.0129
-0.0135    0.0393    0.1503    0.0393   -0.0135
-0.0052   -0.0135   -0.0129   -0.0135   -0.0052
```

$g_h^{[1,0]} =$

```
-0.0036   -0.0116   -0.0000    0.0116    0.0036
-0.0231   -0.0737   -0.0000    0.0737    0.0231
-0.0429   -0.1366   -0.0000    0.1366    0.0429
-0.0231   -0.0737   -0.0000    0.0737    0.0231
-0.0036   -0.0116   -0.0000    0.0116    0.0036
```

$g_h^{[0,1]} =$

```
 0.0036    0.0231    0.0429    0.0231    0.0036
 0.0116    0.0737    0.1366    0.0737    0.0116
-0.0000   -0.0000   -0.0000   -0.0000   -0.0000
-0.0116   -0.0737   -0.1366   -0.0737   -0.0116
-0.0036   -0.0231   -0.0429   -0.0231   -0.0036
```

$g_h^{[1,1]} =$

```
-0.0093   -0.0296   -0.0000    0.0296    0.0093
-0.0296   -0.0944   -0.0000    0.0944    0.0296
-0.0000    0.0000    0.0000   -0.0000   -0.0000
 0.0296    0.0944    0.0000   -0.0944   -0.0296
 0.0093    0.0296    0.0000   -0.0296   -0.0093
```

$g_h^{[2,0]} =$

```
 0.0085    0.0037   -0.0243    0.0037    0.0085
 0.0541    0.0234   -0.1551    0.0234    0.0541
 0.1003    0.0435   -0.2876    0.0435    0.1003
 0.0541    0.0234   -0.1551    0.0234    0.0541
 0.0085    0.0037   -0.0243    0.0037    0.0085
```

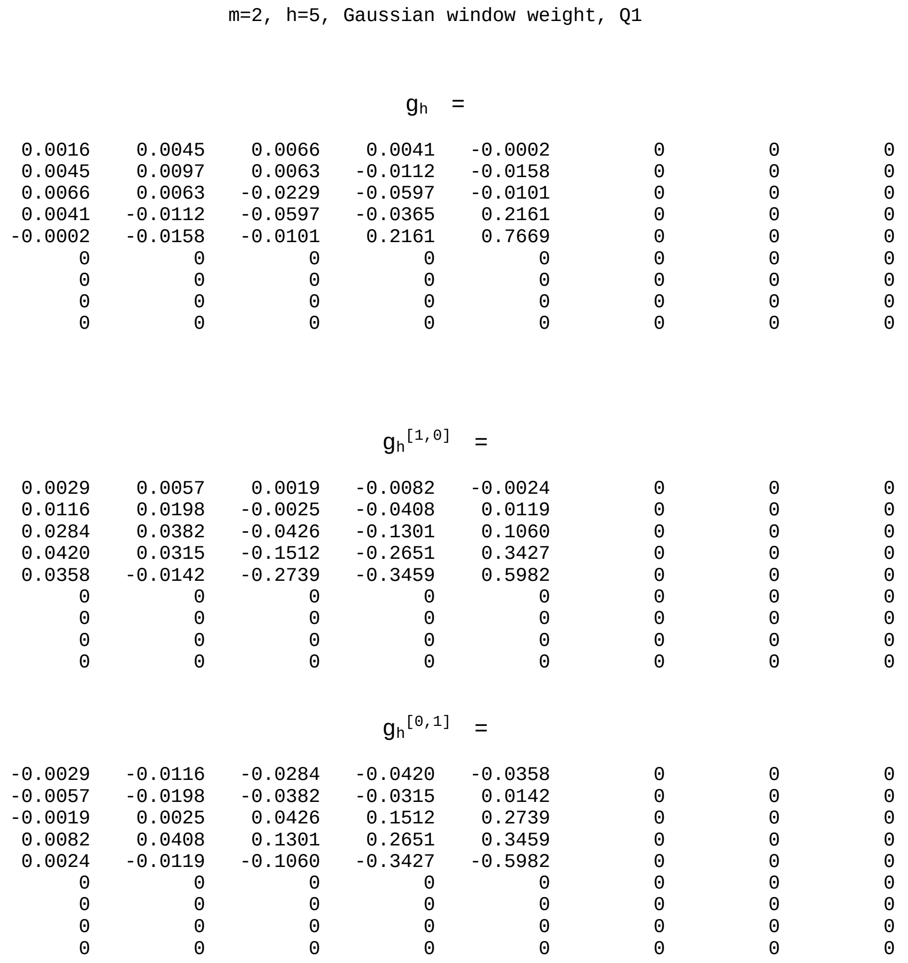
$g_h^{[0,2]} =$

```
 0.0085    0.0541    0.1003    0.0541    0.0085
 0.0037    0.0234    0.0435    0.0234    0.0037
-0.0243   -0.1551   -0.2876   -0.1551   -0.0243
 0.0037    0.0234    0.0435    0.0234    0.0037
 0.0085    0.0541    0.1003    0.0541    0.0085
```

Figure 1.16: Symmetric LPA masks for estimation (smoothing) and differentiation. The LPA window weight w is Gaussian, the power $m = 2$. The mask size is 5×5, $h = 5$. The reference pixel is at the center of the mask.

```
                    m=1, h=5, Gaussian window weight, Q1

                                  gh =

   -0.0007   -0.0028   -0.0065   -0.0086   -0.0058         0         0         0
   -0.0028   -0.0101   -0.0210   -0.0221   -0.0041         0         0         0
   -0.0065   -0.0210   -0.0345   -0.0100    0.0589         0         0         0
   -0.0086   -0.0221   -0.0100    0.0918    0.2537         0         0         0
   -0.0058   -0.0041    0.0589    0.2537    0.4905         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0


                                gh[1,0]  =

   -0.0005   -0.0016   -0.0025   -0.0004    0.0050         0         0         0
   -0.0023   -0.0074   -0.0118   -0.0020    0.0235         0         0         0
-0.0070      -0.0225   -0.0357   -0.0062    0.0714         0         0         0
   -0.0136   -0.0438   -0.0696   -0.0121    0.1390         0         0         0
   -0.0170   -0.0546   -0.0869   -0.0151    0.1736         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0

                                gh[0,1] =

    0.0005    0.0023    0.0070    0.0136    0.0170         0         0         0
    0.0016    0.0074    0.0225    0.0438    0.0546         0         0         0
    0.0025    0.0118    0.0357    0.0696    0.0869         0         0         0
    0.0004    0.0020    0.0062    0.0121    0.0151         0         0         0
   -0.0050   -0.0235   -0.0714   -0.1390   -0.1736         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
```

Figure 1.17: Non-symmetric LPA masks for estimation (smoothing) and differentiation. The LPA window weight w is Gaussian, the first quadrant. The reference (estimation) pixel is at the right down corner of the non-zero mask.



```
                   m=2, h=5, Gaussian window weight, Q1


                                    gh  =

    0.0016    0.0045    0.0066    0.0041   -0.0002         0         0         0
    0.0045    0.0097    0.0063   -0.0112   -0.0158         0         0         0
    0.0066    0.0063   -0.0229   -0.0597   -0.0101         0         0         0
    0.0041   -0.0112   -0.0597   -0.0365    0.2161         0         0         0
   -0.0002   -0.0158   -0.0101    0.2161    0.7669         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0



                                  gh[1,0]  =

    0.0029    0.0057    0.0019   -0.0082   -0.0024         0         0         0
    0.0116    0.0198   -0.0025   -0.0408    0.0119         0         0         0
    0.0284    0.0382   -0.0426   -0.1301    0.1060         0         0         0
    0.0420    0.0315   -0.1512   -0.2651    0.3427         0         0         0
    0.0358   -0.0142   -0.2739   -0.3459    0.5982         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0

                                  gh[0,1]  =

   -0.0029   -0.0116   -0.0284   -0.0420   -0.0358         0         0         0
   -0.0057   -0.0198   -0.0382   -0.0315    0.0142         0         0         0
   -0.0019    0.0025    0.0426    0.1512    0.2739         0         0         0
    0.0082    0.0408    0.1301    0.2651    0.3459         0         0         0
    0.0024   -0.0119   -0.1060   -0.3427   -0.5982         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
```

Figure 1.18: Non-symmetric LPA masks for estimation (smoothing) and differentiation. The LPA window weight w is Gaussian, the first quadrant. The reference (estimation) pixel is right down corner of the non-mask.

```
                    m=2, h=5, Gaussian window weight, Q1
                               (Continuation)


                                  gh[1,1]  =

  -0.0016   -0.0053   -0.0084   -0.0015    0.0167         0         0         0
  -0.0053   -0.0169   -0.0268   -0.0047    0.0536         0         0         0
  -0.0084   -0.0268   -0.0426   -0.0074    0.0851         0         0         0
  -0.0015   -0.0047   -0.0074   -0.0013    0.0148         0         0         0
   0.0167    0.0536    0.0851    0.0148   -0.1702         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0


                                  gh[2,0]  =

   0.0014    0.0019   -0.0024   -0.0068    0.0058         0         0         0
   0.0069    0.0090   -0.0111   -0.0321    0.0274         0         0         0
   0.0208    0.0273   -0.0339   -0.0975    0.0832         0         0         0
   0.0406    0.0531   -0.0659   -0.1899    0.1621         0         0         0
   0.0507    0.0663   -0.0823   -0.2372    0.2025         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0


                                  gh[0,2]  =

   0.0014    0.0069    0.0208    0.0406    0.0507         0         0         0
   0.0019    0.0090    0.0273    0.0531    0.0663         0         0         0
  -0.0024   -0.0111   -0.0339   -0.0659   -0.0823         0         0         0
  -0.0068   -0.0321   -0.0975   -0.1899   -0.2372         0         0         0
   0.0058    0.0274    0.0832    0.1621    0.2025         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0
```

Figure 1.19: Non-symmetric LPA masks for differentiation. The LPA window weight is Gaussian, the first quadrant. The reference (estimation) pixel is left down corner of the non-mask.

1.5 Directional regular grid estimators

1.5.1 Directional LPA

The estimators defined by the formulas (1.40)-(1.42) for $\Delta_1 = \Delta_2 = 1$ have a form

$$\hat{y}_h(x) = \sum_{l_1}\sum_{l_2} g_h(l_1,\ l_2) z(x_1 - l_1,\ x_2 - l_2), \tag{1.87}$$

$$\hat{y}_h^{(k)}(x) = \sum_{l_1}\sum_{l_2} g_h^{(k)}(l_1,\ l_2) z(x_1 - l_1,\ x_2 - l_2), \tag{1.88}$$

where the homogeneous kernels of the estimators are given as

$$g_h(l_1,\ l_2) = w_h(l_1,\ l_2)\phi^T(0,0)\Phi_h^{-1}\phi(l_1,\ l_2), \tag{1.89}$$

$$g_h^{(k)}(l_1,\ l_2) = w_h(l_1,\ l_2)[D_{k_1,k_2}\phi^T(0,0)]\Phi_h^{-1}\phi(l_1,\ l_2), \tag{1.90}$$

$$\Phi_h = \sum_{l_1}\sum_{l_2} w_h(l_1,\ l_2)\phi(l_1,\ l_2)\phi^T(l_1,\ l_2). \tag{1.91}$$

When the polynomial approximation is in use we apply the formulas (1.7) for the items of the vectors $\phi(l_1,\ l_2)$ of the variables l_1 and l_2. These polynomials enable the polynomial fit of the image intensity in the directions of the horizontal and vertical rectangular coordinate axes X, Y.

By default, this standard approximation assumes that the axes X and Y are treated as main directions with the best approximation of the image intensity variations along these directions. In this case the intensity variations in a different direction, say in diagonal of a quadrant, require use of higher order polynomial approximation than it is really necessary. For example, if the intensity is a quadratic polynomial in the diagonal direction it requires only three coefficient approximation. However, the standard $X - Y$ axis fit requires second power polynomials on both variables with the total number of coefficients equal to 6. As a result the minimal size of the kernel used for filtering is increasing.

In order to repair this sort of drawbacks and improve a fitting ability of the basic LPA estimators we introduce novel directional LPA estimators exploiting directional kernels and directional local polynomial fits.

Let us rotate the original $X - Y$ coordinates on the angle θ as

$$\begin{aligned} u(l_1, l_2, \theta) &= l_1 \cos\theta + l_2 \sin\theta, \\ v(l_1, l_2, \theta) &= -l_1 \sin\theta + l_2 \cos\theta, \end{aligned} \tag{1.92}$$

and consider the new coordinates $u(l_1, l_2, \theta)$, $v(l_1, l_2, \theta)$, relative coordinates of observations with respect to the reference point $(x_1,\ x_2)$, as functions of the old ones l_1, l_2 (see Figure 1.20) and the angle θ.

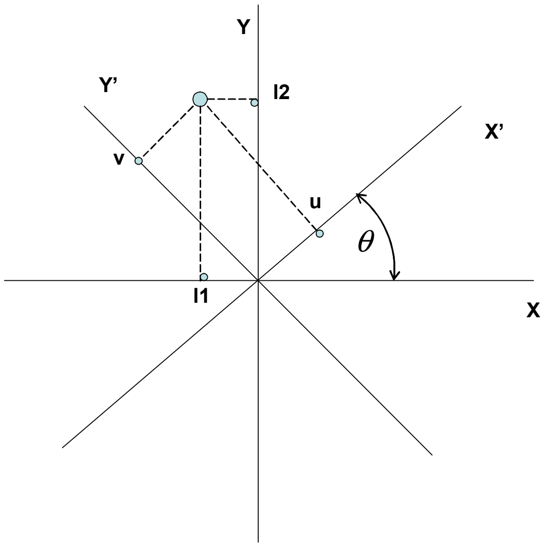


Figure 1.20: An axes rotation for design of the directional kernels.

Then, the directional LPA estimators are

$$\hat{y}_h(x) = \sum_{l_1}\sum_{l_2} \tilde{g}_h(l_1, l_2, \theta) z(x_1 + l_1,\ x_2 + l_2), \tag{1.93}$$

$$\hat{y}_h^{(k)}(x) = \sum_{l_1}\sum_{l_2} \tilde{g}_h^{(k)}(l_1, l_2, \theta) z(x_1 + l_1,\ x_2 + l_2). \tag{1.94}$$

The new directional kernels $\tilde{g}_h(l_1, l_2, \theta)$ and $\tilde{g}_h^{(k)}(l_1, l_2, \theta)$ depend on the rotation angle θ

$$\tilde{g}_h(l_1, l_2, \theta) = g_h(u(l_1, l_2, \theta), v(l_1, l_2, \theta)), \tag{1.95}$$

$$\tilde{g}_h^{[k_1,k_2]}(l_1, l_2, \theta) = g^{(k)}(u(l_1, l_2, \theta),\ v(l_1, l_2, \theta)), \tag{1.96}$$

where g_h and $g_h^{(k)}$ are defined by the equations (1.89)-(1.90).

We use also different powers of the polynomials $m = (m_1,\ m_2)$ and different window sizes $h = (h_1,\ h_2)$ on the variables in order to reinforce the directional properties of the kernels .

Then, we define the kernels by the expressions:

$$\tilde{g}_h(l_1, l_2, \theta) = w(u/h_1, v/h_2) \cdot \phi^T(0,0)\Phi_h^{-1}\phi(u,\ v), \tag{1.97}$$
$$\tilde{g}_h^{[k_1,k_2]}(l_1, l_2, \theta) = w(u/h_1, v/h_2) \cdot [D_{k_1,k_2}\phi^T(0,0)]\Phi_h^{-1}\phi(u,\ v), \tag{1.98}$$
$$\Phi_h = \sum_{l_1}\sum_{l_2} w(u/h_1, v/h_2)\phi(u,\ v)\phi^T(u,\ v), \tag{1.99}$$

where the arguments of u and v given by (1.92) are omitted for simplicity of notation.

The vector $\phi(u,\ v)$ in (1.97)-(1.99) is composed from the complete set of $2D$ polynomials of the powers m_1 and m_2 for the variables u and v respectively.

The transformation (1.92) rotates the coordinates X, Y to X', Y' on the angle θ. In the new coordinates X',Y' the LPA fit used in the formulas (1.97)-(1.99) coincides with used in the previous sections with the only difference that the different powers applied for the arguments corresponding to the axes X', Y'.

Let us ones more illustrate a difference of approximation in the old and new rotated axes (variables). Assume that we use $m_1 = 2$ and $m_2 = 0$ (1.97). The kernel has the minimal acceptable window size 3×1 with the total minimal number of the pixel equal to 3. This kernel is able to yield the accurate reconstruction of the image features which are given as the polynomial in the coordinate X', Y' of the power $m = (2, 0)$.

In order to obtain the same accurate reconstruction of the polynomial image features in the original coordinate system X, Y we need use the LPA of the power $m = (2, 2)$ with the total minimal number of the pixels in the kernel equal to 6. Thus, the rotation enables one to decrease a minimal size of the kernel and the smaller kernel means an ability to recognize and reconstruct details of a smaller size.

Another important aspect of the new kernels is linked with the different sizes h_1 and h_2 used for the different coordinates. It makes possible to give an accurate (say, high power) approximation along edges with small size windows. In this way the filtering can be accurately directed along the edge and it does not involve a wide band of pixels from a neighborhood of the edge.

1.5.2 Numerical examples

We will show some filtering directional kernels. Our main intention is to demonstrate the directivity property as well as to show a design flexibility of the new kernels.

Let

$$W_U(u,v) = 1, \text{ if } |u| \leq 1, \ |v| \leq 1$$

and

$$W_G(u,v) = \exp(-(|u|^2 + |v|^2)/2), \text{ if } |u|^2 + |v|^2 \leq 1$$

be uniform and Gaussian finite supports, respectively.

These basic supports are exploited in order to obtain directional window functions used in the LPA. The following three types of the windows are introduced :

(1) Type 00

$$\begin{aligned} w(u,v) &= W_U(u,v), \\ w(u,v) &= W_G(u,v), \end{aligned} \tag{1.100}$$

(2) Type 10

$$\begin{aligned} w(u,v) &= W_U(u,v) \cdot [u]_+, \\ w(u,v) &= W_G(u,v) \cdot [u]_+, \end{aligned} \tag{1.101}$$

(2) Type 11

$$\begin{aligned} w(u,v) &= W_U(u,v) \cdot [u]_+ \cdot [v]_+, \\ w(u,v) &= W_G(u,v) \cdot [u]_+ \cdot [v]_+. \end{aligned} \tag{1.102}$$

Here $[u]_+$ is an indicator-function, $[u]_+ = 1$ if $u > 0$ and $[u]_+ = 0$ if $u \leq 0$.

The windows of Type 00 are symmetric in the axes X', Y' (see Figure 1.20), i.e. $w(u,v) = w(-u,v)$, $w(u,v) = w(u,-v)$, $w(u,v) = w(-u,-v)$, with the rectangular uniform or ellipsoid Gaussian basic support $W_U(u,v)$ and $W_G(u,v)$, respectively.

The windows of Type 10 are symmetric with respect to Y' and nonsymmetric with respect to X', for example for $u > 0$ and $v > 0$, $w(u,v) = w(u,-v)$, $w(-u,v) = 0$. These windows are zero for $u \leq 0$ and equal to one-half of the corresponding windows of the Type 00.

The windows of Type 11 are nonsymmetric with respect to both coordinates X' and Y'. For $u > 0$ and $v > 0$, $w(-u,v) = 0$, $w(-u,-v) = 0$, $w(u,-v) = 0$. These windows are nonzero for $u > 0$ and $v > 0$ only and they are formed as one-fourth of the corresponding windows of the Type 00.

The windows (1.100)-(1.102) are substituted in the formulas (1.97)-(1.99) giving the LPA kernels for smoothing and differentiation.

Figures 1.21-1.35 demonstrate shapes and directivity of the kernels as a function of the angle θ for different types of the window functions and different orders of the polynomials used in the design.

Uniform white color of the kernel imaging in Figures 1.21 (Type 00) and

1.23 (Type 10) shows that equal weights are used in the approximation by the zero power polynomials with the uniform window function $W_U(u,v)$. However, the zero power LPA results in non-equal values of the corresponding kernels if the Gaussian support $W_G(u,v)$ is used, see Figures 1.28 (Type 00), 1.30 (Type 10) and 1.34 (Type 11).

Effects of the second power polynomials can be observed in Figures 1.22 (Type 00). The kernels form bands of equal values in the direction of the axe Y' perpendicular to the axis X'.

Remind that the use of the first power approximation $m = (1,0)$ gives the kernels coinciding with the kernels of the zero power $m = (0,0)$ if the used window is symmetric with respect the corresponding variable u (axe X'). In a similar way, the kernels of the orders $m = (2,0)$ and $m = (3,0)$ also coincide for the similar symmetric window.

However, for the non-symmetric supports the orders $m = (0,0)$, $(1,0)$, $(2,0)$, $(3,0)$ give different kernels, compare Figures 1.23-1.26 (Type 10).

The Gaussian finite support used in Figures 1.28-1.35 gives the kernels which are different from the rectangular support kernels by the ellipsoid form of the support as well as by non-equal weights even when we use the zero-power polynomial approximations.

Figures 1.21-1.35 demonstrate that Types 01 and Type 11 kernels have supports formed as a one-half or one-fourth of the support of the symmetric Type 00 kernels. However, it does not mean that the kernel, say, of Type 11 is just one-fourth of the corresponding kernel derived for the Type 00 window with the same power of the polynomials.

All kernel weights are shown in the images on symmetric square masks. The areas of these masks beyond the kernel support are filled by zeros shown by black or grey colors. The reference point of the estimate is a center point of this square masks. This representation of the kernels is an important moment as it simplifies calculation of the corresponding $2D$ convolution estimates.

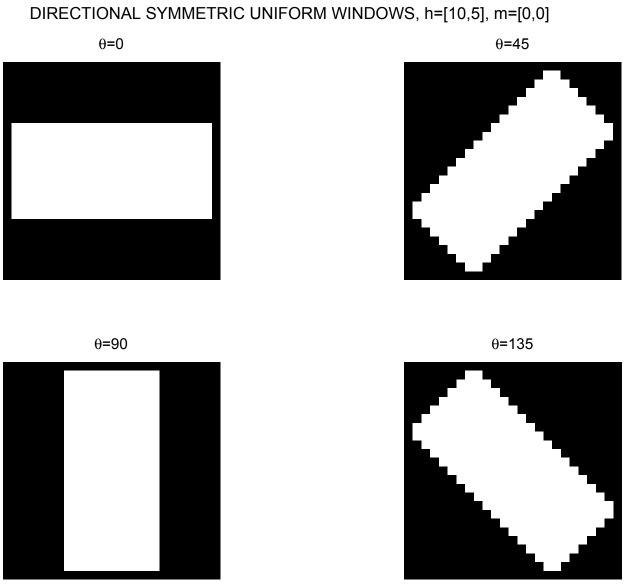


Figure 1.21:

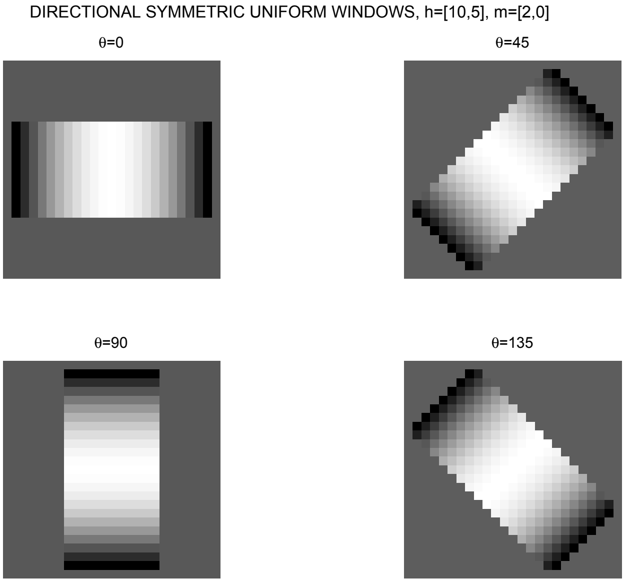


Figure 1.22:

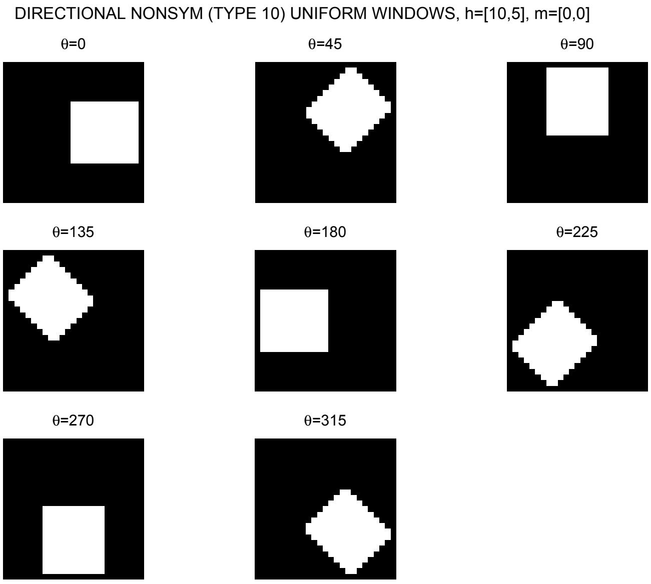


Figure 1.23:

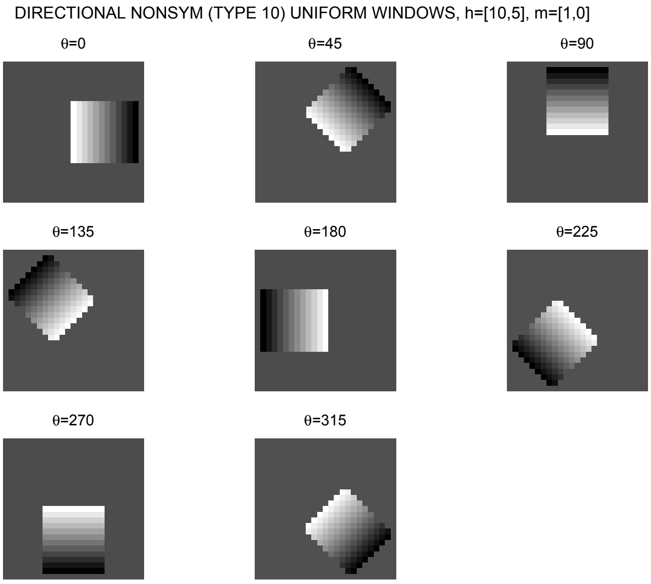


Figure 1.24:

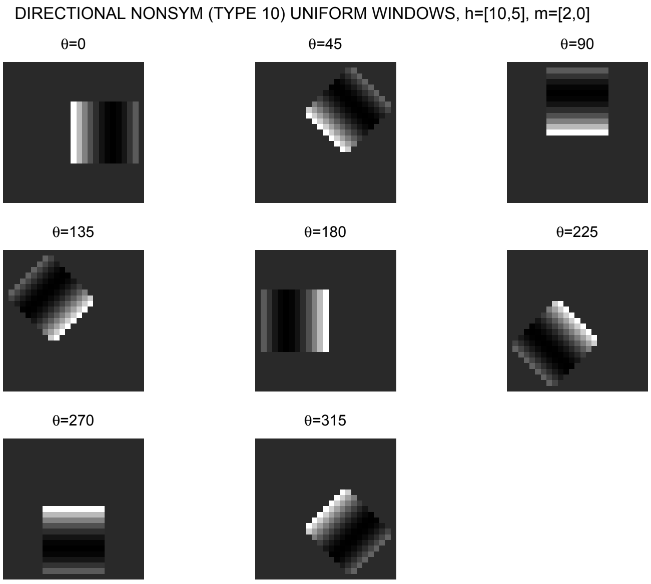

DIRECTIONAL NONSYM (TYPE 10) UNIFORM WINDOWS, h=[10,5], m=[2,0]
θ=0
θ=45
θ=90
θ=135
θ=180
θ=225
θ=270
θ=315

Figure 1.25:

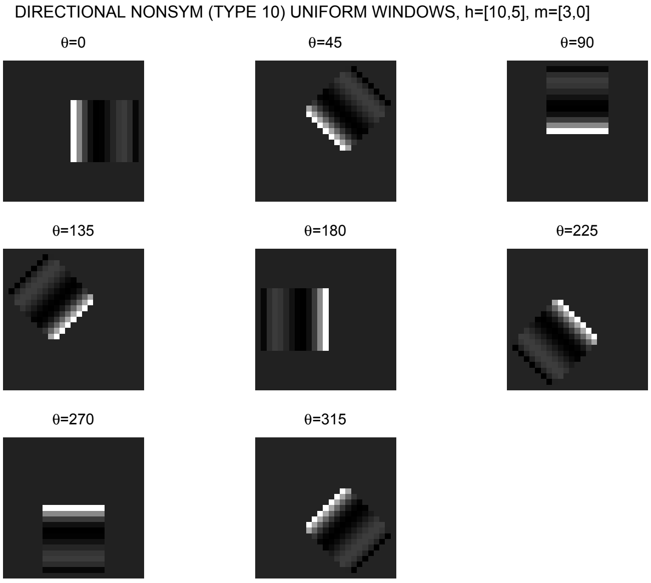

DIRECTIONAL NONSYM (TYPE 10) UNIFORM WINDOWS, h=[10,5], m=[3,0]
θ=0
θ=45
θ=90
θ=135
θ=180
θ=225
θ=270
θ=315

Figure 1.26:

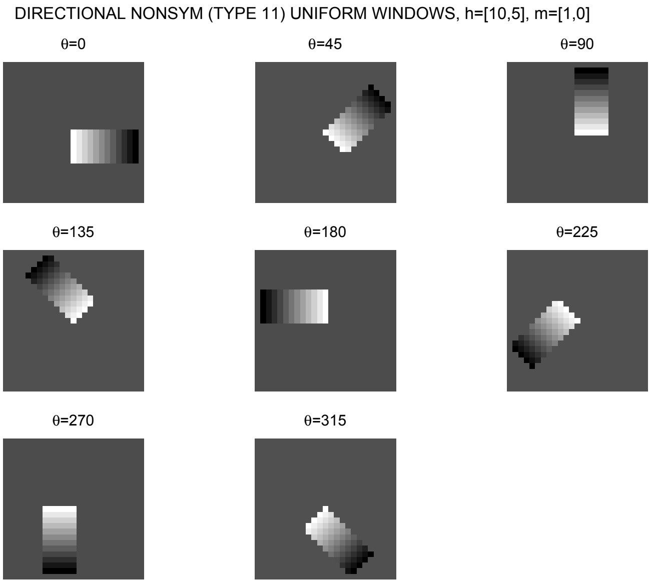


Figure 1.27:

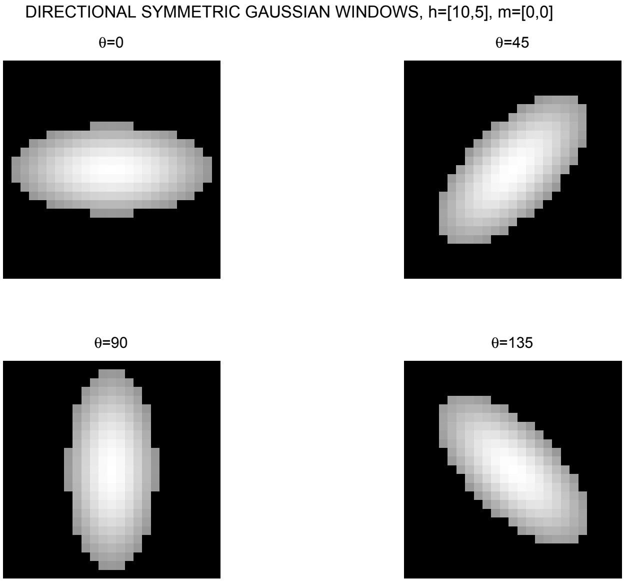


Figure 1.28:

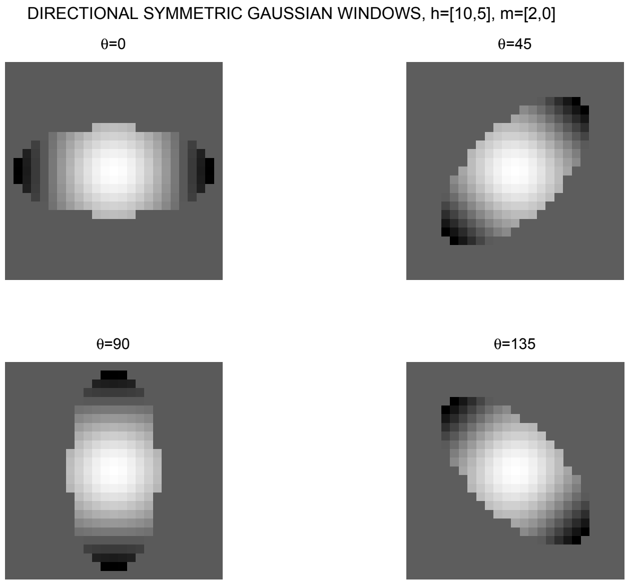


Figure 1.29:

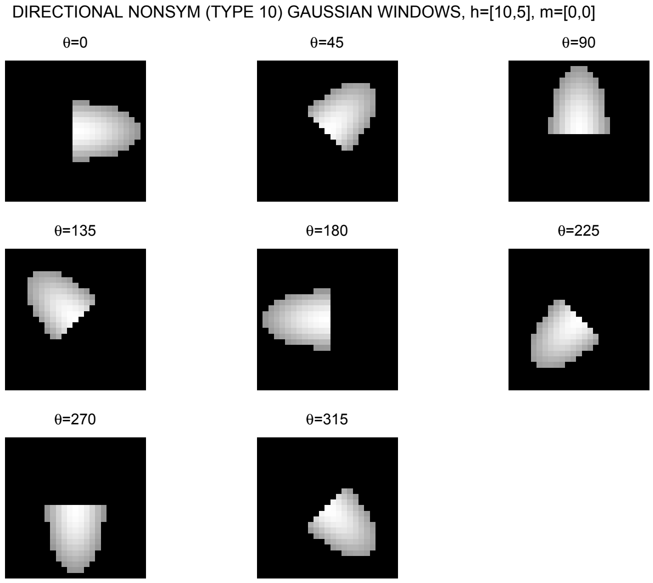

DIRECTIONAL NONSYM (TYPE 10) GAUSSIAN WINDOWS, h=[10,5], m=[0,0]
θ=0
θ=45
θ=90
θ=135
θ=180
θ=225
θ=270
θ=315

Figure 1.30:

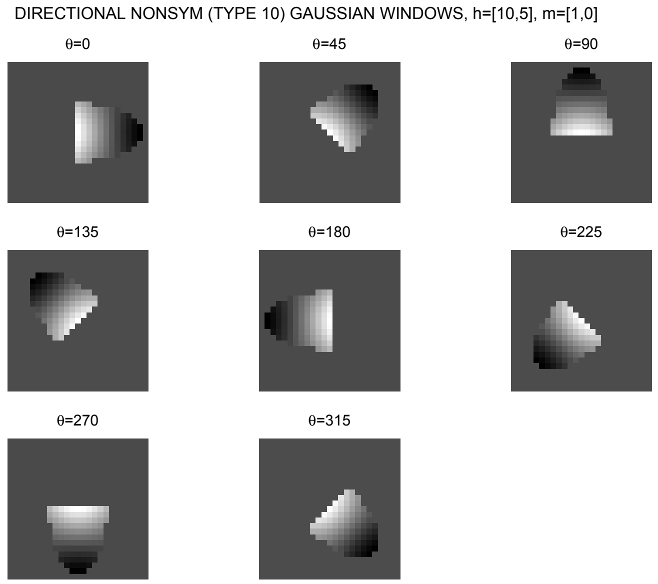


Figure 1.31:

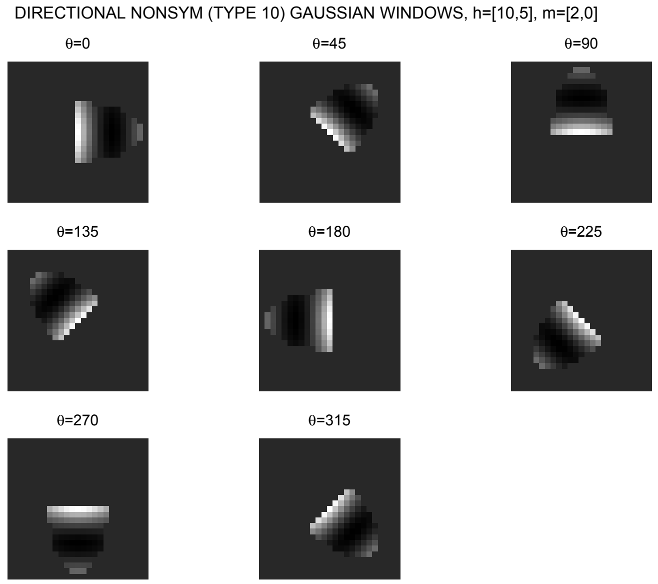


Figure 1.32:

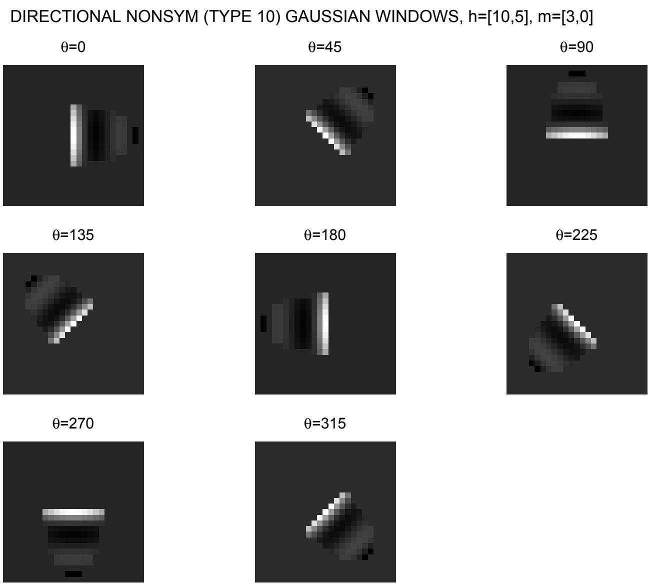


Figure 1.33:

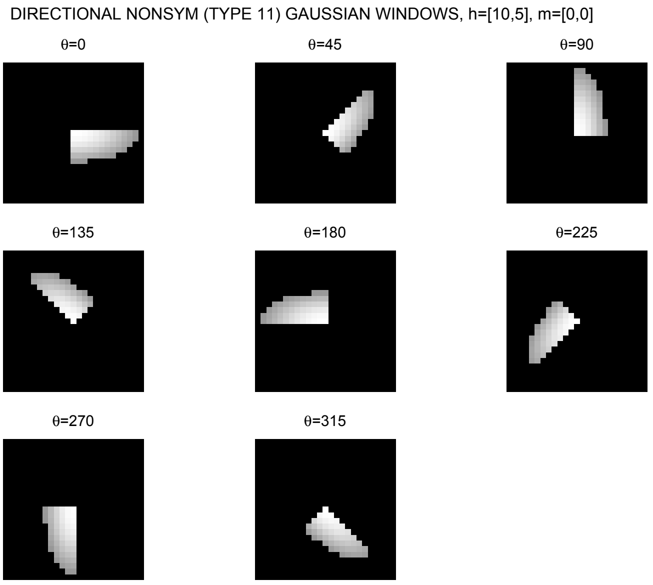


Figure 1.34:

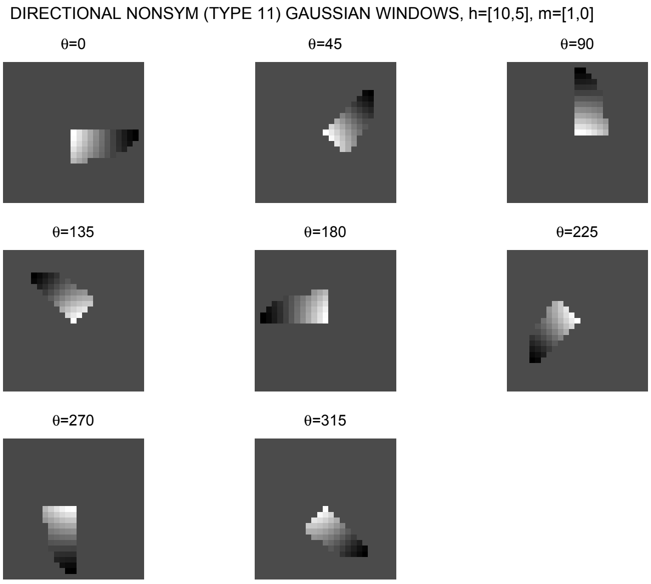

DIRECTIONAL NONSYM (TYPE 11) GAUSSIAN WINDOWS, h=[10,5], m=[1,0]
θ=0
θ=45
θ=90
θ=135
θ=180
θ=225
θ=270
θ=315

Figure 1.35:

1.6 Integral kernel estimators

1.6.1 Integral LPA estimates

An integral representation of linear $2D$ filters applied for image processing is quite standard for the theory and for many applications. At least the integral form makes the filter presentation simpler and many its properties more transparent. There are two ways how these integral estimators can be introduced in the context of the LPA approach.

First, we can start from the regular grid discrete estimators (1.40)-(1.42) and consider their limit representations as the grid size parameters Δ_1 and Δ_2 go to zero. Another way is to introduce the integral version of the LPA, where the sum in the criteria function (1.8) is replaced by the corresponding integrals. Final results are completely identical for the both approaches.

The second way, i.e. the integral version of the LPA approach, was proposed in [55], [56], [58], where integral multivariable smoothing and differentiation operators have been derived and studied.

In this section we use the first approach starting from the already obtained discrete estimators (1.40)-(1.42). The limit results can be finalized in the following compact form.

Proposition 1 *Let the discrete kernel estimates of the function and the derivatives be defined by (1.40)-(1.42), where $\phi(x)$ are polynomials (1.7) and the window w_h is of the form (1.9).*

Then, the limit corresponding estimates, as Δ_1, $\Delta_2 \rightarrow 0$, have the following integral forms

$$\hat{y}_h(x) = \int_{R^2} g(u) z(x - hu) du, \tag{1.103}$$

$$g(x) = w(x)\phi^T(0)\Phi^{-1}\phi(x), \tag{1.104}$$

$$\Phi = \int_{R^2} w(u)\phi(u)\phi^T(u) du \tag{1.105}$$

and

$$\hat{y}_h^{(k)}(x) = \frac{1}{h^k}\int_{R^2} g^{(k)}(u) z(x - hu) du, \tag{1.106}$$

$$g^{(k)}(x) = (-1)^{|k|} w(x)[D^{(k)}\phi^T(0)]\Phi^{-1}\phi(x), \tag{1.107}$$

where the $2D$ integrals are calculated over $2D$ variables u, $\int_{R^2} du = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} du_1 du_2$.

Proof of Proposition 1.

Firstly, derive the formula (1.105) for the matrix Φ_h. Let $T_d(h)$ denote a diagonal matrix composed from the items of the vector $\phi(x)$ with $x_1 = h$, $x_2 = h$, $T_d(h) = diag\{\phi(x)\}$ (1.46).

Successive steps of the proof are as follows

$$
\begin{aligned}
&\Phi_h = \frac{1}{\Delta^2}\sum_{l_1}\sum_{l_2} w(\frac{l_1\Delta}{h}, \frac{l_2\Delta}{h})T_d(h)\phi(\frac{l_1\Delta}{h}, \frac{l_2\Delta}{h}) \times \qquad (1.108)\\
&\phi^T(\frac{l_1\Delta}{h}, \frac{l_2\Delta}{h})T_d(h)\frac{\Delta^2}{h^2} = \\
&\frac{1}{\Delta^2}T_d(h)\sum_{l_1}\sum_{l_2} w(\frac{l_1\Delta}{h}, \frac{l_2\Delta}{h})\phi(\frac{l_1\Delta}{h}, \frac{l_2\Delta}{h})\phi^T(\frac{l_1\Delta}{h}, \frac{l_2\Delta}{h})\frac{\Delta^2}{h^2}T_d(h) \to \\
&\frac{1}{\Delta^2}T_d(h)\cdot \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w(u_1,u_2)\phi(u_1,u_2)\phi^T(u_1,u_2)du_1du_2 \cdot T_d(h)\\
= \; &\frac{1}{\Delta^2}T_d(h)\Phi T_d(h), \qquad (1.109)\\
&\Phi = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w(u_1,u_2)\phi(u_1,u_2)\phi^T(u_1,u_2)du_1du_2,
\end{aligned}
$$

where the integration variables u_1 and u_2 are obtained as $l_1\Delta/h \to u_1$, $l_2\Delta/h \to u_2$ with $du_1 = \Delta/h$, $du_2 = \Delta/h$. It proofs (1.105), where compact notation for the $2D$ integration is used.

Substituting the last formula (1.109) for Φ_h in (1.40) we obtain

$$
\begin{aligned}
&\hat{y}_h(x) = \sum_{l_1}\sum_{l_2} g_h(l_1\Delta_1, l_2\Delta_2)z(x_1 - l_1\Delta_1, x_2 - l_2\Delta_2) = \\
&\frac{1}{\Delta^2}\sum_{l_1}\sum_{l_2} w(\frac{l_1\Delta_1}{h}, \frac{l_2\Delta_2}{h})\phi^T(0)\Phi_h^{-1}T_d(h)\phi(\frac{l_1\Delta_1}{h}, \frac{l_2\Delta_2}{h}) \times \\
&z(x_1 - h\frac{l_1\Delta_1}{h}, x_2 - h\frac{l_2\Delta_2}{h})\frac{\Delta^2}{h^2} = \\
&\sum_{l_1}\sum_{l_2} w(\frac{l_1\Delta_1}{h}, \frac{l_2\Delta_2}{h})\phi^T(0)T_d^{-1}(h)\Phi^{-1}T_d^{-1}(h)T_d(h)\phi(\frac{l_1\Delta_1}{h}, \frac{l_2\Delta_2}{h}) \times \\
&z(x_1 - h\frac{l_1\Delta_1}{h}, x_2 - h\frac{l_2\Delta_2}{h})\frac{\Delta^2}{h^2} = \\
\to \; &\int\int w(u_1,u_2)\phi^T(0)T_d^{-1}(h)\Phi^{-1}\phi(u_1,u_2)z(x_1 - hu_1, x_2 - hu_2)du_1du_2,
\end{aligned}
$$

what follows by

$$
\begin{aligned}
&\hat{y}_h(x) = \int_{R^2} g_h(u)z(x - hu)du, \qquad (1.110)\\
&g_h(x) = w(x)\phi^T(0)T_d^{-1}(h)\Phi^{-1}\phi(x),\\
&\Phi = \int_{R^2} w(u)\phi(u)\phi^T(u)du, \qquad (1.111)
\end{aligned}
$$

It can be seen because of (1.7) that

$$\phi^T(0)T_d^{-1}(h) = \phi^T(0). \tag{1.112}$$

Then (1.103) follows from (1.110).

In a similar way, substituting (1.109) for Φ_h in (1.42) we obtain for the derivative estimate

$$\begin{aligned} \hat{y}_h^{(k)}(x) &= \int_{R^2} g_h^{(k)}(u) z(x - hu) du, \\ g_h^{(k)}(x) &= (-1)^{|k|} w(x) [D^{(k)} \phi^T(0)] T_d^{-1}(h) \Phi^{-1} \phi(x). \end{aligned} \tag{1.113}$$

It can be seen that for the polynomial $\phi(x)$ (1.7)

$$\begin{aligned} g_h^{(k)}(x) &= \frac{1}{h^{k_1+k_2}} g^{(k)}(x), \\ g^{(k)}(x) &= (-1)^{|k|} w(x) [D^{(k)} \phi^T(0)] \Phi^{-1} \phi(x). \end{aligned} \tag{1.114}$$

Inserting the last formula in (1.113) proves the equation (1.106). It completes the proof of Proposition 1.

In order to be more accurate, we note that some extra conditions on the window $w(x)$ and the intensity $z(x)$ are required in order the limit passages of the proposition have a place. However, we do not go to these mathematical details as these conditions are quite not restrictive. Actually, the existence of the integrals in (1.110)-(1.113) is a sufficient assumption for the all limit passages.

Proposition 1 shows that the kernel (1.104) of the function estimation does not depend on h, while the kernel of the differentiation (1.107) is an exponential function of h, having the exponent $1/h^{k_1+k_2}$ as a factor of the kernel independent on h.

Consider the *reproductive properties* of the integral estimators.

Let the input signal $y(x)$ be a vector equal to $\phi(x)$ then the formulas (1.103) and (1.106) give the accurate values of the polynomials and derivatives:

$$\begin{aligned} &\int_{R^2} g(u) \phi(x - hu) du = \phi(x), \\ &\frac{1}{h^k} \int_{R^2} g^{(k)}(u) \phi(x - hu) du = D^{(k)} \phi(x). \end{aligned}$$

These equations are identities on x.

Then for $x = 0$

$$\begin{aligned}&\int_{R^2} g(u)\phi(hu)du = \phi(0), \qquad (1.115)\\ &\frac{1}{h^k}\int_{R^2} g^{(k)}(u)\phi(-hu)du = D^{(k)}\phi(0).\end{aligned}$$

Note that $\phi(hu) = T_d(h)\phi(u)$ and inserting the formulas for $g(u)$ and $g^{(k)}(u)$ from (1.103) and (1.106) in (1.115) we find that the latter equations are simplified to

$$\begin{aligned}&\int_{R^2} g(u)T_d(h)\phi(u)du = \phi(0),\\ &\int_{R^2} \frac{1}{h^k}g^{(k)}(u)T_d(h)\phi(-u)du = D^{(k)}\phi(0),\end{aligned}$$

which results in

$$\begin{aligned}&\int_{R^2} g(u)\phi(u)du = \phi(0), \qquad (1.116)\\ &\int_{R^2} g^{(k)}(u)\phi(u)du = (-1)^{|k|}D^{(k)}\phi(0),\end{aligned}$$

or

$$\begin{aligned}&\int\int g(u_1,u_2)u_1^{r_1}u_2^{r_2}du_1u_2 = \delta_{r_1,0}\delta_{r_2,0}, \qquad (1.117)\\ &\int\int g^{(k)}(u_1,u_2)\frac{(u_1)^{r_1}}{r_1!}\frac{(u_2)^{r_2}}{r_2!}du_1u_2 = (-1)^{r_1+r_2}\delta_{r_1,k_1}\delta_{r_2,k_2},\\ &r_1+r_2 \leq m,\end{aligned}$$

where g and $g^{(k)}$ are defined in (1.112) and (1.114) and $\delta_{r,k}$ denotes the Kronecker symbol.

Changing the integration variables in (1.103) and (1.106) gives two quite useful equivalent forms of the kernel estimates:

(a)

$$\hat{y}_h(x) = \frac{1}{h^2}\int_{R^2} g(\frac{u}{h})y(x-u)du, \qquad (1.118)$$

$$\hat{y}_h^{(k)}(x) = \frac{1}{h^2}\int_{R^2} g^{(k)}(\frac{u}{h})y(x-u)du, \qquad (1.119)$$

(b)

$$\hat{y}_h(x) = \frac{1}{h^2}\int_{R^2} g(\frac{x-u}{h})y(u)du, \tag{1.120}$$

$$\hat{y}_h^{(k)}(x) = \frac{1}{h^2}\int_{R^2} g^{(k)}(\frac{x-u}{h})y(u)du, \tag{1.121}$$

The difference is, say between (1.118) and (1.120), that the dependence on x in (1.118) analytically goes from input $y(x)$ and in (1.120) from the kernel $g(x)$. The similar difference is identified between the representations (1.119) and (1.121).

1.6.2 Analytical examples

In this section we use the classical orthogonal polynomials (Legendre, Chebyshev, and Hermit) instead of the polynomials in the standard form (1.7). The technique is completely similar to exploited in Section 1.4.1 for the analytical design of the discrete smoothing and differentiation estimators.

Let the $2D$ polynomials $\phi(x)$ be orthogonal with the window $w(x)$. Then, the matrix Φ (1.105) is identical , $\Phi = I$, and the kernel formulas (1.112) and (1.114) are simplified to the form

$$g(x) = w(x)\phi^T(0)\phi(x), \tag{1.122}$$

and

$$g^{(k)}(x) = (-1)^{|k|}w(x)[D^{(k)}\phi^T(0)]\phi(x). \tag{1.123}$$

Assume that the $2D$ window $w(x)$ is a product of $1D$ windows $w_1(x_1)$ and $w_1(x_2)$

$$w(x) = w_1(x_1)w_1(x_2). \tag{1.124}$$

Then, the $2D$ polynomials $\phi(x)$ orthogonal with the window $w(x)$ (1.124) can be composed from the $1D$ polynomials orthogonal with the weights $w_1(x_1)$ and $w_1(x_2)$, respectively (see formulas (1.77)). In this case the kernels (1.122) and (1.123) can be derived using these $1D$ orthogonal polynomials. It is exactly as it was done in Section 1.4.1 in order to obtain the analytical formulas for the kernels of the discrete estimators.

(I) **1D Legendre polynomials** orthogonal with the symmetric window function

$$w_1(x) = 1,\ |x| \leq 1$$

have the form

$$\phi_1(x) = \sqrt{\frac{1}{2}},\ \phi_2(x) = \sqrt{\frac{3}{2}}x,\ \phi_3(x) = \frac{1}{2}\sqrt{\frac{5}{2}}[3x^2-1). \tag{1.125}$$

The corresponding smoothing kernels of the powers $m = 0, 1, 2$ follow from (1.122)

$$g(x) = \left\{ \begin{array}{c} w(x) \text{ for } m = 0, 1, \\ [\frac{7}{2} - \frac{15}{4}(x_1^2 + x_2^2)]w(x) \text{ for } m = 2, \end{array} \right. \tag{1.126}$$

where

$$w(x) = \frac{1}{4} w_1(x_1) w_1(x_2).$$

The first derivative estimation kernels follow from (1.123) are as follows

$$\begin{array}{rcl} g^{(1,0)}(x) & = & -3x_1 w(x), \text{ for } m = 1, 2, \\ g^{(0,1)}(x) & = & -3x_2 w(x), \text{ for } m = 1, 2. \end{array} \tag{1.127}$$

The second derivative estimation kernels follow from (1.123) are as follows

$$\begin{array}{rcl} g^{(1,1)}(x) & = & 9x_1 x_2 w(x), \text{ for } m = 2, \\ g^{(2,0)}(x) & = & \frac{15}{2}(3x_1^2 - 1)w(x), \text{ for } m = 2, \\ g^{(0,2)}(x) & = & \frac{15}{2}(3x_2^2 - 1)w(x), \text{ for } m = 2. \end{array}$$

(II) **1D Legendre polynomials** orthogonal with the nonsymmetric window function

$$w_1(x) = 1, \ 0 \le x \le 1$$

have the form

$$\phi_1(x) = 1, \ \phi_2(x) = \sqrt{3}(2x - 1), \ \phi_3(x) = \frac{1}{2}\sqrt{5}[3(2x - 1)^2 - 1].$$

Let

$$w(x) = w_1(x_1) w_1(x_2).$$

The corresponding smoothing kernels follow from (1.122)

$$g(x) = \left\{ \begin{array}{l} w(x) \text{ for } m = 0, \\ [7 - 6(x_1 + x_2)]w(x) \text{ for } m = 1, \\ [21 + 30(x_1^2 + x_2^2) - 54(x_1 + x_2) + 36x_1 x_2]w(x) \text{ for } m = 2. \end{array} \right.$$

The first derivative estimation kernels follow from (1.123) are as follows

$$g^{(1,0)}(x) = \left\{ \begin{array}{l} -6(2x_1 - 1)w(x) \text{ for } m = 1, \\ -[12(2x_1 - 1)x_2 - \frac{15}{2}(3(2x_1 - 1)^2 - 1)]w(x) \text{ for } m = 2. \end{array} \right.$$

$$g^{(0,1)}(x) = \begin{cases} -6(2x_2-1)w(x) \text{ for } m=1, \\ -[12(2x_2-1)x_1 - \frac{15}{2}(3(2x_2-1)^2-1)]w(x) \text{ for } m=2. \end{cases}$$

The second derivative estimation kernels follow from (1.123) are as follows

$$\begin{aligned} g^{(1,1)}(x) &= 36(2x_1-1)(2x_2-1)w(x), \text{ for } m=2, \\ g^{(2,0)}(x) &= 15[3(2x_1-1)^2-1]w(x), \text{ for } m=2, \\ g^{(0,2)}(x) &= 15[3(2x_2-1)^2-1]w(x), \text{ for } m=2. \end{aligned}$$

(III) **1D Laguerre polynomials** orthogonal with the nonsymmetric window function

$$w_1(x) = \exp(-x),\ 0 \le x < \infty$$

have the form

$$\phi_1(x) = 1,\ \phi_2(x) = (1-x),\ \phi_3(x) = \frac{1}{2}(2-4x+x^2).$$

Let

$$w(x) = w_1(x_1)w_1(x_2).$$

The corresponding smoothing kernels follow from (1.122)

$$g(x) = \begin{cases} w(x) \text{ for } m=0, \\ [3-(x_1+x_2)]w(x) \text{ for } m=1, \\ [6+\frac{1}{2}(x_1^2+x_2^2) - 4(x_1+x_2) + x_1x_2]w(x) \text{ for } m=2. \end{cases}$$

The first derivative estimation kernels follow from (1.123) are as follows

$$g^{(1,0)}(x) = \begin{cases} -(x_1-1)w(x) \text{ for } m=1, \\ [(1-x_1)(3-x_1-x_2)+2-4u_1+x_1^2]w(x) \text{ for } m=2. \end{cases}$$

$$g^{(0,1)}(x) = \begin{cases} -(x_2-1)w(x) \text{ for } m=1, \\ [(1-x_2)(3-x_1-x_2)+2-4u_2+x_2^2]w(x) \text{ for } m=2. \end{cases}$$

The second derivative estimation kernels follow from (1.123) are as follows

$$\begin{aligned} g^{(1,1)}(x) &= (1-x_1)(1-x_2)w(x), \text{ for } m=2, \\ g^{(2,0)}(x) &= \frac{1}{2}(2-4u_1+x_1^2)w(x), \text{ for } m=2, \\ g^{(0,2)}(x) &= \frac{1}{2}(2-4u_2+x_2^2)w(x),, \text{ for } m=2. \end{aligned}$$

(IV) **Chebyshev polynomials** orthogonal with the symmetric window function

$$w_1(x) = \sqrt{1-x^2},\ |x| \leq 1$$

have the form

$$\phi_1(x) = \sqrt{\frac{2}{\pi}},\ \phi_2(x) = \sqrt{\frac{2}{\pi}}x,\ \phi_3(x) = \sqrt{\frac{2}{\pi}}[4x^2 - 1).$$

The corresponding smoothing kernels of the powers $m = 0, 1, 2$ follow from (1.122)

$$g(x) = \begin{cases} w(x) \text{ for } m = 0, 1, \\ [3 - 4(x_1^2 + x_2^2)]w(x) \text{ for } m = 2, \end{cases}$$

where

$$w(x) = \frac{4}{\pi^2} w_1(x_1) w_1(x_2).$$

The first derivative estimation kernels follow from (1.123) are as follows

$$\begin{aligned} g^{(1,0)}(x) &= -4u_1 w(x),\ \text{for } m = 1, 2, \\ g^{(0,1)}(x) &= -4u_2 w(x),\ \text{for } m = 1, 2. \end{aligned}$$

The second derivative estimation kernels follow from (1.123) are as follows

$$\begin{aligned} g^{(1,1)}(x) &= (1 - x_1)(1 - x_2)w(x),\ \text{for } m = 2, \\ g^{(2,0)}(x) &= \frac{1}{2}(2 - 4x_1 + x_1^2)w(x),\ \text{for } m = 2, \\ g^{(0,2)}(x) &= \frac{1}{2}(2 - 4x_2 + x_2^2)w(x),\ \text{for } m = 2. \end{aligned}$$

(V) **Hermitian polynomials** orthogonal with the symmetric window function

$$w_1(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2),\ -\infty < x < \infty,$$

have the form

$$\phi_1(x) = 1,\ \phi_2(x) = x,\ \phi_3(x) = \frac{1}{\sqrt{2}}(x^2 - 1).$$

The corresponding smoothing kernels of the powers $m = 0, 1, 2$ follow from (1.122)

$$g(x) = \begin{cases} w(x) \text{ for } m = 0, 1, \\ [2 - \frac{1}{2}(x_1^2 + x_2^2)]w(x) \text{ for } m = 2, \end{cases} \tag{1.128}$$

where

$$w(x) = \frac{1}{\pi} w_1(x_1) w_1(x_2).$$

The first derivative estimation kernels follow from (1.123) are as follows

$$\begin{aligned} g^{(1,0)}(x) &= -x_1 w(x), \text{ for } m = 1, 2, \\ g^{(0,1)}(x) &= -x_2 w(x), \text{ for } m = 1, 2. \end{aligned} \tag{1.129}$$

The second derivative estimation kernels follow from (1.123) are as follows

$$\begin{aligned} g^{(1,1)}(x) &= x_1 x_2 w(x), \text{ for } m = 2, \\ g^{(2,0)}(x) &= (x_1^2 - 1)w(x), \text{ for } m = 2, \\ g^{(0,2)}(x) &= (x_2^2 - 1)w(x), \text{ for } m = 2. \end{aligned} \tag{1.130}$$

1.6.3 Links with the generalized function theory

Let us consider the integral estimate (1.110) as a smoother of the input signal $y(x)$

$$\hat{y}_h(x) = \int_{R^2} g(u) y(x - hu) du, \tag{1.131}$$

with the Gaussian kernel (1.128) for $m = 0$

$$g(x) = \frac{1}{2\pi} \exp(-\frac{1}{2}||x||^2). \tag{1.132}$$

Two equivalent forms (1.118) and (1.120) for the Gaussian kernel give:

(a)

$$\begin{gathered} \hat{y}_h(x) = \frac{1}{h^2} \int_{R^2} g(u/h) y(x - u) du = \\ \frac{1}{2\pi h^2} \int_{R^2} \exp(-\frac{1}{2}||u||^2/h^2) y(x - u) du, \end{gathered} \tag{1.133}$$

(b)

$$\hat{y}_h(x) = \frac{1}{h^2} \int_{R^2} g((x-u)/h) y(u) du = \tag{1.134}$$
$$\frac{1}{2\pi h^2} \int_{R^2} \exp(-\frac{1}{2}||x-u||^2/h^2) y(u) du.$$

The following facts can be stated for these $\hat{y}_h(x)$ and $g(x/h)/h^2$:

(1) The smoothed $\hat{y}_h(x)$ is continuous for any $h \geq 0$ even for the discontinuous original $y(x)$;

(2) The smoothed $\hat{y}_h(x)$ is infinitely differentiable regardless properties of $y(x)$;

(3) If $y(x)$ is continuous at the point x then

$$\lim_{h \to 0} \hat{y}_h(x) = y(x);$$

(4) If $y(x)$ has a finite jump at the point x then

$$\lim_{h \to 0} \hat{y}_h(x) = \frac{1}{2}(y(x+0) + y(x-0)),$$

where $y(x+0)$ and $y(x-0)$ are values of $y(x)$ at the left and right hand sides from the jump-point x;

(5) As $h \to 0$ the kernel $g(x/h)/h^2$ approaches the Dirac delta function

$$\lim_{h \to 0} g(x/h)/h^2 = \delta(x). \tag{1.135}$$

The delta function is a unusual singular function. To include mathematical objects such as the Dirac delta function $\delta(x)$ into analysis, the concept of a function should be somehow extended. It is done in terms of the so-called generalized functions.

Let us present some notes on the mathematical definition of the generalized functions.

Consider $g(x/h)/h^2$ as a sequence function with a parameter h. Note that $g(x/h)/h^2$ is a usual function for any $h > 0$. Then on the definition "generalized function" is the limit of this sequence function as $h \to 0$, i.e.

$$\delta(x) = \lim_{h \to 0} g(x/h)/h^2. \tag{1.136}$$

Important to emphasize that this limit delta function $\delta(x)$ is very different from the usual $g(x/h)/h^2$, $h > 0$, and mathematically strictly defined with use special linear functionals (convolutions with smooth "test" kernels).

The estimate $\hat{y}_h(x)$ is also considered as a sequence function with a parameter h. The corresponding "generalized function" (in this case not the delta function) is the limit of the sequence $\hat{y}_h(x)$ as $h \to 0$, i.e.

$$\hat{y}_0(x) = \lim_{h\to 0} \hat{y}_h(x). \tag{1.137}$$

It deserves to be noted that this limit $\hat{y}_0(x)$ of the smoothed function $\hat{y}_h(x)$ can be not identical to the original function $y(x)$.

In general, the properties of the generalized functions are very different from the properties of the ordinary functions. In particular, the generalized functions have derivatives of all orders.

In engineering contexts, the functional nature of the delta function is often suppressed and $\delta(x)$ is instead viewed as a "special kind" of function, resulting in the useful notation $\delta(x)$. However, the concept of the delta function as a limit of some sequence functions is in common use.

Go further and consider some derivative estimates which follow from the integral representation for smoothing estimators (1.118) and (1.120).

Differentiate the both sides of (1.120). Then

$$\frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}}\hat{y}_h(x) = \frac{1}{h^2}\int_{R^2}\frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}} g(\frac{x-u}{h})y(u)du = \tag{1.138}$$

$$\frac{1}{h^{2+k_1+k_2}}\int_{R^2} g^{(k)}(\frac{x-u}{h})y(u)du, \tag{1.139}$$

where it is denoted

$$g^{(k)}(x) = \frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}} g(x). \tag{1.140}$$

Change the integration variables in the transform (1.139). Then

$$\frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}}\hat{y}_h(x) = \frac{1}{h^{k_1+k_2}}\int_{R^2} g^{(k)}(u)y(x-hu)du. \tag{1.141}$$

Note also that it follows from (1.131) that

$$\frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}}\hat{y}_h(x) = \int_{R^2} g(u)\frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}} y(x-hu)du = \tag{1.142}$$

$$\frac{1}{h^2}\int_{R^2} g(\frac{x-u}{h})\frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}} y(u)du.$$

This last formula allows one to make a number of interesting conclusions.

The above properties of the smoothing estimator of $\hat{y}_h(x)$ are applicable to the derivative $\frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}}\hat{y}_h(x)$:

(1) The derivative $\frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}}\hat{y}_h(x)$ is continuous for any $h \geq 0$ even for discontinuous $\frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}} y(x)$;

(2) The derivative $\frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}}\hat{y}_h(x)$ is infinitely differentiable regardless properties of $\frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}}y(x)$;

The derivative $\frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}}\hat{y}_h(x)$ is a sequence function defining the limit generalized function as $\lim_{h\to 0}\frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}}\hat{y}_h(x)$.

(3) If the derivative $\frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}}y(x)$ is continuous at the point x then

$$\lim_{h\to 0}\frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}}\hat{y}_h(x)=\frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}}y(x). \tag{1.143}$$

(4) As $h\to 0$ the kernel $\frac{1}{h^{2+k_1+k_2}}g^{(k)}(\frac{x-u}{h})$ approaches the derivative of the Dirac delta function

$$\lim_{h\to 0}\frac{1}{h^{2+k_1+k_2}}g^{(k)}(\frac{x}{h})=\frac{\partial^{k_1+k_2}}{\partial x_1^{k_1}\partial x_2^{k_2}}\delta(x). \tag{1.144}$$

The equation (1.143) is of importance as it shows that the differentiation of (1.118) and (1.120) can be used in order to obtain the derivative estimators. Even more, it is proved that this derivative estimator is a convolution of $y(x)$ with the kernel $g^{(k)}$ explicitly defined by (1.140) as the derivative of the smoothing kernel $g(x)$.

We specially apply the notation $g^{(k)}$ for this derivative the same as used earlier, say in (1.123), for the weights of the differentiators in order to emphasize this meaning of these weights.

Thus, the formula (1.140) gives the kernels of the differentiators. The differentiation of the smoothing kernel $g(x)$ is a simple and constructive way to design the differentiation kernels.

Note that in general (for general $w(x)$) the result for $g^{(k)}$ can be different when we use the formula (1.140) and the regular LPA technique. However, for the Gaussian $w(x)$ the results are identical.

Let us produce some calculations and derive the derivative estimators according to the formula (1.143) for the Gaussian $g(x)$ (1.132).

It can be seen for the derivatives of $g(x)$ that

$$\begin{aligned}
&\frac{\partial}{\partial x_1}g(x)=-x_1g(x),\ \frac{\partial}{\partial x_2}g(x)=-x_2g(x),\\
&\frac{\partial^2}{\partial x_1\partial x_2}g(x)=x_1x_2g(x),\ \frac{\partial^2}{\partial x_1^2}g(x)=(x_1^2-1)g(x),\\
&\frac{\partial^2}{\partial x_2^2}g(x)=(x_2^2-1)g(x).
\end{aligned}$$

Then, the derivatives kernels following from (1.140) are in the form

$$\begin{aligned} g^{(1,0)}(x) &= -x_1 g(x),\ g^{(0,1)}(x) = -x_2 g(x), \\ g^{(1,1)}(x) &= x_1 x_2 g(x),\ g^{(2,0)}(x) = (x_1^2 - 1) g(x), \\ g^{(0,2)}(x) &= (x_2^2 - 1) g(x). \end{aligned} \tag{1.145}$$

Comparing (1.145) with the formulas given by the LPA (1.129)-(1.130) we can see that these results are identical.

It can be mentioned that

$$\begin{gathered} g_{Lap}(x) = g^{(0,2)}(x) + g^{(2,0)}(x) = \\ (x_1^2 + x_2^2 - 2) g(x) \end{gathered} \tag{1.146}$$

gives the kernel for the Gaussian based estimate of the Laplacian of $y(x)$, i.e. the estimate of

$$\frac{\partial^2}{\partial x_1^2} y(x) + \frac{\partial^2}{\partial x_2^2} y(x).$$

In conclusion of the section we wish to remark that in the real function theory and in the distribution theory the generalized functions is a special subject. The rigorous mathematical treatment of the field can be seen in the multivolume work by Gel'fand and Shilov [24]-[25] and in the book by Friedlander [23].

1.7 LPA accuracy

The window size is a crucial point in the efficiency of the local estimators (e.g. [20], [59], [83]). When the window size h is relatively small, the LPA gives a good smooth fit of functions but then fewer data are used and the estimates are more variable and sensitive with respect to the noise. The best choice of h involves a trade-off between the bias and variance, which depends on the order of the derivatives being involved in the LPA, a sample period, the noise variance and the derivatives of $y(x)$ beyond the power used in the LPA. We present the accuracy analysis of the LPA estimates in order to illustrate these statements. For simplicity we consider only the homogeneous estimators while the similar results can be obtained for the general case of nonhomogeneous LPA estimates.

First, we derive the formulas for the discrete estimates with finite, small or large, values of h. As a next step we consider the asymptotic accuracy errors assuming that h is small. It allows to obtain a clear analytical formulas, which are used for window size optimization.

1.7.1 Discrete estimators

The estimation error is defined as a difference between the estimated signal and the estimate

$$\begin{aligned} e_y(x,h) &= y(x) - \hat{y}_h(x), \\ e_y^{(k)}(x,h) &= y^{(k)}(x) - \hat{y}_h^{(k)}(x). \end{aligned}$$

For the observation model (1.1) these errors are composed from the systematic (bias) and random components corresponding to the deterministic $y(x)$ and the random noise $\varepsilon(x)$ respectively:

$$\begin{aligned} m_y(x,h) &= E\{y(x) - \hat{y}_h(x)\} = y(x) - E\{\hat{y}_h(x)\}, \\ m_{y^{(k)}}(x,h) &= E\{y^{(k)}(x) - \hat{y}_h^{(k)}(x)\} = y^{(k)}(x) - E\{\hat{y}_h^{(k)}(x)\}, \end{aligned}$$

where $E\{\}$ means the mathematical expectation.

For the homogeneous convolution estimates defined by the formulas (1.29) and (1.28) respectively for the function

$$\begin{aligned} \hat{y}_h(x) &= \sum_s g_h(u_s) z(x-u_s), \qquad (1.147) \\ g_h(x) &= w_h(x)\phi^T(0)\Phi_h^{-1}\phi(x). \end{aligned}$$

and the derivative

$$\hat{y}_h^{(k)}(x) = \sum_s g_h^{(k)}(u_s) z(x - u_s), \tag{1.148}$$
$$g_h^{(k)}(x) = (-1)^{|k|} w_h(x)[D^{(k)} \phi^T(0)] \Phi_h^{-1} \phi(x),$$
$$\Phi_h = \sum_s w_h(u_s) \phi(u_s) \phi^T(u_s),$$

these bias errors are of the form

$$m_y(x, h) = y(x) - \sum_s g_h(u_s) y(x - u_s), \tag{1.149a}$$
$$m_{y^{(k)}}(x, h) = y^{(k)}(x) - \sum_s g_h^{(k)}(u_s) y(x - u_s). \tag{1.149b}$$

Assume that:

(H1) The image intensity $y(x)$ be deterministic locally smooth belonging to a nonparametric class of continuous $r-$differentiable functions (1.4)

$$F_r(\bar{L}_r) = \{ \max_{r_1+r_2=r} \left| D^{(r)} y(x) \right| = L_r(x) \leq \bar{L}_r, \ \forall \ r_1 + r_2 = r \}, \tag{1.150}$$

where $D^{(r)} = \frac{\partial^{r_1+r_2}}{\partial x_1^{r_1} \partial x_2^{r_2}}$ is a differentiation operator and $\bar{L}_r$ is a finite constant;

(H2) The LPA estimates of the intensity and the derivatives are defined by the equations (1.147) and (1.148). The power of the LPA is equal to m.

Proposition 2 *Let the hypotheses H1 and H2 hold,* $M = \min\{m + 1, |r|\}$, *and the noise in the observation model (1.1) be white with the variance* σ^2. *Then the accuracy of the estimates (1.147) and (1.148) are defined by the following upper bounds for the bias*

$$|m_y(x, h)| \leq \quad \bar{L}_M \sum_{r_1+r_2=M} \frac{1}{r_1! r_2!} \sum_s |g_h(u_s)| |u_{1,s}|^{r_1} |u_{2,s}|^{r_2}, \tag{1.151}$$
$$|m_{y^{(k)}}(x, h)| \leq \quad \bar{L}_M \sum_{r_1+r_2=M} \frac{1}{r_1! r_2!} \sum_s |g_h^{(k)}(u_s)| |u_{1,s}|^{r_1} |u_{2,s}|^{r_2} \tag{1.152}$$

and the following formulas for the variance

$$\sigma_y^2(x, h) = \sigma^2 \sum_s (g_h(u_s))^2, \tag{1.153}$$
$$\sigma_{y^{(k)}}^2(x, h) = \sigma^2 \sum_s (g_h^{(k)}(u_s))^2. \tag{1.154}$$

Proof of Proposition 2.

Consider the systematic error estimation.

If $y(x) \in \mathcal{F}_r(\bar{L}_r)$ (1.4) then, provided

$$M + 1 \leq r,$$

the Taylor series for $y(x - u_s)$ can be used in the form

$$y(x-u_s) = \sum_{k=0}^{M} \sum_{r_1+r_2=k} \frac{(-1)^{r_1+r_2}}{r_1!r_2!} u_{1,s}^{r_1} u_{2,s}^{r_2} D^{(r_1,r_2)} y(x) + \tag{1.155}$$
$$\sum_{r_1+r_2=M+1} \frac{(-1)^{r_1+r_2}}{r_1!r_2!} u_{1,s}^{r_1} u_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda_s u_s),$$
$$0 \leq \lambda_s \leq 1,$$

where the last remainder term is given in the Lagrange form and according to (1.4)

$$\max_{r_1+r_2=M+1} |D^{(r_1,r_2)} y(x - \lambda_s u_s)| \leq \bar{L}_{M+1}. \tag{1.156}$$

For the power m, the LPA kernels (1.147) and (1.148) have , the following reproductive properties (1.27):

$$\sum_s g_h(u_s) u_{1,s}^{k_1} u_{2,s}^{k_2} = \delta_{k_1,0} \cdot \delta_{k_2,0}, \tag{1.157}$$

$$\sum_s g_h^{(k)}(u_s) \frac{(-1)^{r_1+r_2}}{r_1!r_2!} u_{1,s}^{r_1} u_{2,s}^{r_2} = \frac{1}{h^{r_1+r_2}} \delta_{r_1,k_1} \cdot \delta_{r_2,k_2}, \tag{1.158}$$
$$r_1 + r_2 \leq m,$$

Firstly, consider the case $r \geq m + 1$.

Then, we can use (1.155) with $M = m$. Inserting this Taylor series, $M = m$, in (1.147) and using (1.157) we obtain the estimate expectation in the form

$$E\{\hat{y}_h(x)\} = \sum_s g_h(u_s) y(x - u_s) =$$
$$y(x) + \sum_s g_h(u_s) \cdot \sum_{r_1+r_2=m+1} \frac{(-1)^{r_1+r_2}}{r_1!r_2!} u_{1,s}^{r_1} u_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda u_s).$$

Then, the bias of the estimation error is as follows

$$m_e(x,h) = y(x) - E\{\hat{y}_h(x)\} =$$
$$-\sum_s g_h(u_s) \cdot \sum_{r_1+r_2=m+1} \frac{(-1)^{r_1+r_2}}{r_1!r_2!} u_{1,s}^{r_1} u_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda u_s)$$

with the upper bound of this error given in the form

$$\begin{gathered} |m_e(x,h)| = |y(x) - E\{\hat{y}_h(x)\}| = \\ |\sum_s g_h(u_s) \cdot \sum_{r_1+r_2=m+1} \frac{1}{r_1!r_2!} u_{1,s}^{r_1} u_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda u_s)| \leq \\ \bar{L}_{m+1} \sum_{r_1+r_2=m+1} \frac{1}{r_1!r_2!} \sum_s |g_h(u_s)||u_{1,s}|^{r_1} |u_{2,s}|^{r_2}, \end{gathered} \tag{1.159}$$

where (1.156) is used.

In a similar way, substituting (1.155) in the estimate of the derivative (1.148) and using the moment equations (1.158) we have for the expectation of this estimate

$$\begin{gathered} E\{\hat{y}_h^{(k)}(x)\} = \sum_s g_h^{(k)}(u_s) y(x - u_s) = \\ y_h^{(k)}(x) + \sum_s g_h^{(k)}(u_s) \cdot \sum_{r_1+r_2=m+1} \frac{(-1)^{r_1+r_2}}{r_1!r_2!} u_{1,s}^{r_1} u_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda_s u_s) \end{gathered}$$

and for the bias error

$$\begin{gathered} m_e^{(k)}(x,h) = y_h^{(k)}(x) - E\{\hat{y}_h^{(k)}(x)\} = \\ -\sum_s g_h^{(k)}(u_s) \cdot \sum_{r_1+r_2=m+1} \frac{(-1)^{r_1+r_2}}{r_1!r_2!} u_{1,s}^{r_1} u_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda_s u_s). \end{gathered} \tag{1.160}$$

The upper bound error for the derivative estimation is

$$\begin{gathered} |m_e^{k_1,k_2}(x,h)| = |y_h^{(k)}(x) - E\{\hat{y}_h^{(k)}(x)\}| = \\ |\sum_s g_h^{(k)}(u_s) \cdot \sum_{r_1+r_2=m+1} \frac{1}{r_1!r_2!} u_{1,s}^{r_1} u_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda_s u_s)| \quad \leq \\ \bar{L}_{m+1} \sum_{r_1+r_2=m+1} \frac{1}{r_1!r_2!} \sum_s |g_h^{(k)}(u_s)||u_{1,s}|^{r_1} |u_{2,s}|^{r_2}. \end{gathered} \tag{1.161}$$

Secondly, consider the another case $r < m+1$. Then, we are able to use (1.155) with $M = r-1$ only. Inserting this Taylor series, $M = r-1$, in the formula for the derivative estimate expectation and using (1.157) we obtain the estimate expectation in the form

$$\begin{gathered} E\{\hat{y}_h(x)\} = \sum_s g_h(u_s) y(x - u_s) = \\ y(x) + \sum_s g_h(u_s) \cdot \sum_{r_1+r_2=r} \frac{(-1)^{r_1+r_2}}{r_1!r_2!} u_{1,s}^{r_1} u_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda u_s). \end{gathered}$$

Then, the formulas for the bias error and its upper bound are as follows

$$m_e(x,h) = y(x) - E\{\hat{y}_h(x)\} = \\ -\sum_s g_h(u_s) \cdot \sum_{r_1+r_2=r} \frac{(-1)^{r_1+r_2}}{r_1!r_2!} u_{1,s}^{r_1} u_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda u_s)$$

$$|m_e(x,h)| = |y(x) - E\{\hat{y}_h(x)\}| = \tag{1.162}$$

$$|\sum_s g_h(u_s) \cdot \sum_{r_1+r_2=m+1} \frac{(-1)^{r_1+r_2}}{r_1!r_2!} u_{1,s}^{r_1} u_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda u_s)| \leq \\ \bar{L}_r \sum_{r_1+r_2=r} \frac{1}{r_1!r_2!} \sum_s |g_h(u_s)||u_{1,s}|^{r_1}|u_{2,s}|^{r_2}.$$

In a similar way, for the derivative estimate we yield

$$E\{\hat{y}_h^{(k)}(x)\} = \sum_s g_h^{(k)}(u_s) y(x - u_s) = \\ y_h^{(k)}(x) + \sum_s g_h^{(k)}(u_s) \cdot \sum_{r_1+r_2=r} \frac{(-1)^{r_1+r_2}}{r_1!r_2!} u_{1,s}^{r_1} u_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda_s u_s)$$

and for the estimation bias error

$$m_e^{(k)}(x,h) = y_h^{(k)}(x) - E\{\hat{y}_h^{(k)}(x)\} = \\ -\sum_s g_h^{(k)}(u_s) \cdot \sum_{r_1+r_2=r} \frac{(-1)^{r_1+r_2}}{r_1!r_2!} u_{1,s}^{r_1} u_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda_s u_s).$$

The upper bound error for the derivative estimation is

$$|m_e^{(k)}(x,h)| = |y_h^{(k)}(x) - E\{\hat{y}_h^{(k)}(x)\}| = \tag{1.163}$$

$$|\sum_s g_h^{(k)}(u_s) \cdot \sum_{r_1+r_2=m+1} \frac{(-1)^{r_1+r_2}}{r_1!r_2!} u_{1,s}^{r_1} u_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda_s u_s)| \leq \\ \bar{L}_r \sum_{r_1+r_2=r} \frac{1}{r_1!r_2!} \sum_s |g_h^{(k)}(u_s)||u_{1,s}|^{r_1}|u_{2,s}|^{r_2}.$$

Comparing the formulas (1.159) versus (1.162) we see that $m+1$ in (1.159) is replaced by r in (1.162). Noting that $M = \min(m+1, r)$ we can see that the result (1.151) of the proposition is a combination of the formulas (1.159) and (1.162). In the same way the result (1.152) of the proposition is a combination of the formulas (1.161) and (1.163).

For the white noise the variance of the estimates are given by the obvious formulas

$$\sigma_y^2(x,h) = \sigma^2 \sum_s (g_h(u_s))^2, \tag{1.164}$$

$$\sigma_{y^{(k)}}^2(x,h) = \sigma^2 \sum_s (g_h^{(k)}(u_s))^2. \tag{1.165}$$

It proves the formulas (1.153)-(1.154) and completes the proof of Proposition 2.

1.7.2 Integral estimators

We consider the asymptotic formulas for the estimation errors provided that the grid is given by (1.2) with $\Delta_1 = \Delta_2 = \Delta$ and Δ small. These results are applied for scale optimization and further analysis. The limit passage $\Delta \to 0$ can be produced in the above equations derived for the discrete estimators. However, it is straightforward to derive these results from the integral estimator representations using the corresponding limit estimates (1.103) and (1.106):

$$\begin{aligned} \hat{y}_h(x) &= \int_{R^2} g(u) z(x - hu) du, \\ g(x) &= w(x)\phi^T(0)\Phi^{-1}\phi(x), \\ \Phi &= \int_{R^2} w(u)\phi(u)\phi^T(u)du, \end{aligned} \tag{1.166}$$

and

$$\begin{aligned} \hat{y}_h^{(k)}(x) &= \frac{1}{h^k}\int_{R^2} g^{(k)}(u) z(x - hu) du, \\ g^{(k)}(x) &= (-1)^{|k|} w(x)[D^{(k)}\phi^T(0)]\Phi^{-1}\phi(x). \end{aligned} \tag{1.167}$$

Let use present result which is an integral version of given in Proposition 2 for the discrete LPA estimates.

Assume that:

(H1) The image intensity $y(x)$ be deterministic locally smooth belonging to a nonparametric class of continuous $r-$differentiable functions (1.4)

$$F_r(\bar{L}_r) = \{ \max_{r_1+r_2=r} \left| D^{(r_1,r_2)} y(x) \right| = L_r(x) \le \bar{L}_r,\ \forall\ r_1 + r_2 = r\},$$

where $D^{(r_1,r_2)} = \frac{\partial^{r_1+r_2}}{\partial x_1^{r_1}\partial x_2^{r_2}}$ is a differentiation operator and $\bar{L}_r$ is a finite constant;

(H2) The integral LPA estimates of the intensity and the derivatives are defined by the equations (1.166) and (1.167). The power of the LPA is equal to m.

Proposition 3 *Let the hypotheses H1 and H2 hold, $M = \min\{m+1, |r|\}$, and the noise in the observation model (1.1) be white with the variance σ^2 and $h \to 0$, $\Delta \to 0$, $h/\Delta \to \infty$. Then the accuracy of the estimates (1.166) and (1.167) are defined by the following upper bounds for the estimation bias*

$$|m_y(x,h)| \leq h^M \bar{L}_M A_e, \qquad (1.168)$$

$$A_e = \sum_{r_1+r_2=M} \frac{1}{r_1! r_2!} \int\int |g(u_1,u_2) u_1^{r_1} u_2^{r_2}| du_1 du_2,$$

$$|m_{y^{(k)}}(x,h)| \leq h^{M-(k_1+k_2)} \bar{L}_M A_e^{(k)}, \qquad (1.169)$$

$$A_e^{(k)} = \sum_{r_1+r_2=M} \frac{1}{r_1! r_2!} \int\int |g^{(k)}(u_1,u_2) u_1^{r_1} u_2^{r_2}| du_1 du_2$$

and the following formulas for the variance

$$\frac{1}{\sigma^2}\sigma_y^2(x,h) \to \frac{\Delta^2}{h^2} B_e, \ B_e = \int\int [g(u_1,u_2)]^2 du_1 du_2, \qquad (1.170)$$

$$\frac{1}{\sigma^2}\sigma_{y^{(k)}}^2(x,h) \to \frac{\Delta^2}{h^{2+2|k|}} B_e^{(k)}, \qquad (1.171)$$

$$B_e^{(k)} = \int\int [g^{(k)}(u_1,u_2)]^2 du_1 du_2.$$

Proof of Proposition 3.

Consider the bias error of estimation. As the basic ideas of the calculations are quite similar to produced for the discrete estimates we omit details.

Taking into consideration the moment equations (1.116), (1.117) and using the Taylor series (1.155) with the power M we obtain the following upper bound for the deterministic systematic errors:

$$\begin{aligned} |m_e(x,h)| = \ & |y(x) - E\{\hat{y}_h(x)\}| = \sum_{r_1+r_2=M} \frac{h^M}{r_1! r_2!} \times \qquad (1.172) \\ & \int\int |g(u_1,u_2) u_1^{r_1} u_2^{r_2} D^{(r_1,r_2)} y(x - \lambda h u_s)| du_1 du_2 \leq \\ & h^M \bar{L}_M A_e, \end{aligned}$$

where

$$A_e = \sum_{r_1+r_2=M} \frac{1}{r_1! r_2!} \int\int |g(u_1,u_2) u_1^{r_1} u_2^{r_2}| du_1 du_2, \qquad (1.173)$$

and

$$
\begin{aligned}
|m_e^{(k)}(x,h)| = \; & |y^{(k)}(x) - E\{\hat{y}_h^{(k)}(x)\}| = \frac{1}{h^{|k|}} \sum_{r_1+r_2=M} \frac{h^M}{r_1! r_2!} \times \\
& \int\int |g^{(k)}(u_1,u_2) u_1^{r_1} u_2^{r_2} D^{(r_1,r_2)} y(x - \lambda h u_s)| du_1 du_2 \leq \\
& h^{M-|k|} \bar{L}_M A_e^{(k)},
\end{aligned} \tag{1.174}
$$

where

$$
A_e^{(k)} = \sum_{r_1+r_2=M} \frac{1}{r_1! r_2!} \int\int |g^{(k)}(u_1,u_2) u_1^{r_1} u_2^{r_2}| du_1 du_2. \tag{1.175}
$$

It proves the formulas (1.168)-(1.169).

The variance of the discrete estimates (1.40) and (1.42) is of the form

$$
\frac{1}{\sigma^2}\sigma_y^2(x,h) = \sum_{l_1}\sum_{l_2} [g_h(l_1\Delta_1,\; l_2\Delta_2)]^2, \tag{1.176}
$$

$$
\frac{1}{\sigma^2}\sigma_{y^{(k)}}^2(x,h) = \sum_{l_1}\sum_{l_2} [g_h^{(k)}(l_1\Delta_1,\; l_2\Delta_2)]^2. \tag{1.177}
$$

The corresponding limits as Δ → 0 can be derived as it was done above for the kernels of the integral estimators. These calculations give compact formulas with a clear indication of the dependence of the variance on the parameters Δ and h.

Let us show this derivation for the variance (1.176). Substitute in (1.176) the expression for $g_h(l_1\Delta_1,\; l_2\Delta_2)$ (1.40) and replace the matrix Φ_h in it by its limit value (1.109).

Then

$$
\begin{aligned}
& g_h(l_1\Delta_1, l_2\Delta_2) = w_h(l_1\Delta_1,\; l_2\Delta_2)\phi^T(0)\Phi_h^{-1}\phi(l_1\Delta_1,\; l_2\Delta_2) = \\
& \frac{\Delta^2}{h^2} w(\frac{l_1\Delta_1}{h},\; \frac{l_2\Delta_2}{h})\phi^T(0)T_d^{-1}(h)\Phi^{-1}\phi(\frac{l_1\Delta_1}{h},\; \frac{l_2\Delta_2}{h}),
\end{aligned}
$$

where T_d is defined in (1.46) and Φ is given by the integral (1.105).

It follows

$$
\begin{aligned}
& \sum_{l_1}\sum_{l_2} [g_h(l_1\Delta_1,\; l_2\Delta_2)]^2 \rightarrow \frac{\Delta^2}{h^2} B_e(h),\; B_e(h) = \int\int [g_h(u_1,u_2)]^2 du_1 du_2, \\
& g_h(u_1,u_2) = w(u_1,u_2)\phi^T(0)T_d^{-1}(h)\Phi^{-1}\phi(u_1,u_2),
\end{aligned}
$$

and as $\phi^T(0)T_d^{-1}(h) = \phi^T(0)$,

$$g_h(u_1, u_2) = g(u_1, u_2) = w(u_1, u_2)\phi^T(0)\Phi^{-1}\phi(u_1, u_2),$$
$$B_e(h) = B_e = \int\int [g(u_1, u_2)]^2 du_1 du_2.$$

Thus, it is follows

$$\frac{1}{\sigma^2}\sigma_y^2(x, h) \to \frac{\Delta^2}{h^2} B_e \text{ as } \Delta \to 0 \tag{1.178}$$

and proves the formula (1.170).

Completely similarly we derive that

$$\frac{1}{\sigma^2}\sigma^2_{y^{(k)}}(x, h) \to \frac{\Delta^2}{h^2} B_e^{(k)}(h), \tag{1.179}$$
$$B_e^{(k)}(h) = \int\int g_h^{(k)}(u_1, u_2)^2 du_1 du_2,$$
$$g_h^{(k)}(u_1, u_2) = w(u_1, u_2)[D^{(k)}\phi^T(0)]T_d^{-1}(h)\Phi^{-1}\phi(u_1, u_2).$$

However, for the polynomials (1.7) the only non-zero items of the vector $[D^{(k)}\phi^T(0)]T_d^{-1}(h)$ is defined according to the formula

$$[D^{(k)}\phi^T(0)]T_d^{-1}(h) = \frac{1}{h^{|k|}}(0, ..., 0, \underbrace{1}_{kth}, 0, ..., 0),$$

where the kth elements of the vector-row corresponds to the location of the polynomial

$$\frac{x_1^{k_1}}{k_1!}\frac{x_2^{k_2}}{k_2!}$$

in the vector $\phi(x)$ used in the LPA.

Then, finally

$$g_h^{(k)}(u_1, u_2) = \frac{1}{h^{k_1+k_2}} g^{(k)}(u_1, u_2),$$
$$g^{(k)}(u_1, u_2) = w(u_1, u_2)[D^{(k)}\phi^T(0)]\Phi^{-1}\phi(u_1, u_2),$$
$$B_e^{(k)}(h) = \frac{1}{h^{2(k_1+k_2)}} B_e^{(k)}, \tag{1.180}$$
$$B_e^{(k)} = \int\int [g^{(k)}(u_1, u_2)]^2 du_1 du_2. \tag{1.181}$$

Substituting (1.180) in (1.179) we prove (1.170).

It completes the proof of Proposition 3.

The special feature of non-asymptotic Proposition 3 in comparison with asymptotic Proposition 2 is that the latter shows clearly a dependence of the bias and the variance on the scale parameter h. Actually it is one of the main reasons for us to derive this sort of asymptotic results.

1.7.3 Mean squared error minimization

Function estimation

The formulas derived in Proposition 3 can be summarized for the considered asymptotic in the compact form:

$$\begin{aligned} |m_e(x,h)| \leq & \quad h^M L_M(x) A_e = |\bar{m}_e(x,h)|, \\ & \sigma_y^2(x,h) = \sigma^2 \frac{\Delta^2}{h^2} B_e \end{aligned} \tag{1.182}$$

Thus, the point-wise mean squared risk $r(x,h)$ for the function in the asymptotic with a small Δ can be represented as follows

$$\begin{aligned} & r(x,h) \triangleq E(y(x) - \hat{y}_h(x))^2 \leq |\bar{m}_e(x,h)|^2 + \sigma_y^2(x,h) = \\ & \bar{r}(x,h) = (h^M L_M(x) A_e)^2 + \frac{\Delta^2 \sigma^2}{h^2} B_e. \end{aligned} \tag{1.183}$$

This upper bound risk function $\bar{r}(x,h)$ is convex on h. Its minimization on h gives the "ideal" values of the scale found from the equation:

$$\frac{\partial}{\partial h} \bar{r}(x,h) = 0.$$

Let us present general auxiliary formulas for further calculation .

Assume that the function to be minimized is of the form:

$$f(h) = \frac{A}{h^a} + B h^{-b}. \tag{1.184}$$

Then the equation $\frac{\partial}{\partial h} f(x) = 0$ gives

$$\begin{aligned} h^* &= (\frac{Bb}{Aa})^{1/(a+b)}, \\ f^* &= f(h^*) = \frac{1}{(h^*)^b} B(1+\gamma^2), \ \gamma^2 = \frac{b}{a}. \end{aligned} \tag{1.185}$$

Comparison (1.183) with (1.184) shows that

$$A = (L_M(x) A_e)^2, \ B = \Delta^2 \sigma^2 B_e, \ a = 2M, \ b = 2$$

and using the formulas (1.185) we obtain:

$$h^*(x) = \left(\Delta^2 \frac{\sigma^2 B_e}{A_e^2 L_M^2(x)} \gamma^2 \right)^{1/(2M+2)}, \ \gamma^2 = \frac{1}{M} \tag{1.186}$$

and

$$\begin{aligned}
&\bar{r}^*(x) = \bar{r}(x, h^*(x)) = \sigma_y^2(x, h^*(x))(1+\gamma^2), && (1.187)\\
&\gamma = |\bar{m}_e(x, h^*(x))|/\sigma_y(x, h^*(x)),\\
&\gamma = \frac{1}{\sqrt{M}} && (1.188)
\end{aligned}$$

Here the parameter γ is introduced as a ratio of the upper bound bias to the standard deviation calculated for the ideal scale $h^*(x)$. The calculation shows that this parameter is a constant depending only on $M = \min\{m+1, |r|\}$, i.e. it depends only on the power of the LPA and the smoothness of the function $y(x)$ defined by the value of r.

The formulas (1.186) and (1.187) demonstrate that the ideal scale $h^*(x)$ depends on the $M-th$ derivatives of $y(x)$ and the ideal variance-bias trade-off is achieved when the ratio between the bias and standard deviation γ is equal to $\frac{1}{\sqrt{M+1}}$. It can be seen also that

$$|\bar{m}_e(x,h)| = \left\{ \begin{array}{l} < \gamma \cdot \sigma_y(x,h) \text{ if } h < h^*(x) \\ > \gamma \cdot \sigma_y(x,h) \text{ if } h > h^*(x). \end{array} \right. . \qquad (1.189)$$

In what follows this inequality is used in order to test the hypotheses: $h \lesseqgtr h^*(x)$ in order to select the scale close to the ideal one.

Derivative estimation

The formulas derived in Proposition 3 for the derivative estimation can be summarized in the compact form:

$$\begin{aligned}
|m_e^{(k)}(x,h)| \leq \ & h^{M-k} L_M(x) A_e^{(k)} = |\bar{m}_e^{(k)}(x,h)|, && (1.190)\\
& \sigma^2_{y^{(k)}}(x,h) = \sigma^2 \frac{\Delta^2}{h^{2+2|k|}} B_e^{(k)},\\
& k = (k_1, k_2),\ |k| = k_1 + k_2.
\end{aligned}$$

Thus, the point-wise mean squared risk $r_e^{(k)}(x,h)$ for the derivative estimation Δ can be represented as follows

$$\begin{aligned}
& r_e^{(k)}(x,h) \triangleq E(D^{(k)}y(x) - \hat{y}_h^{(k)}(x))^2 \leq |m_e^{(k)}(x,h)|^2 + \sigma^2_{y^{(k)}}(x,h),\\
& \bar{r}_e^{(k)}(x,h) = (h^{M-k} L_M(x) A_e^{(k)})^2 + \sigma^2 \frac{\Delta^2}{h^{2+2|k|}} B_e^{(k)}. && (1.191)
\end{aligned}$$

Formally, the analysis of the derivative estimation accuracy is only slightly different from produced for the function estimation. However, the results are different in both the formulas for the ideal scale as well as for the risk functions.

Comparison (1.190) and (1.184) shows that

$$A = (L_M(x)A_e^{(k)})^2,\ B = \Delta^2\sigma^2 B_e^{(k)},\ a = 2(M-k),\ b = 2(1+|k|).$$

and using the formulas (1.185) we obtain for minimization of $\bar{r}_e^{(k)}(x,h)$ (1.191):

$$h_k^*(x) = \left(\Delta^2 \frac{\sigma^2 B_e^{(k)}}{(L_M(x)A_e^{(k)})^2}\gamma^2\right)^{1/(2M+2)},\ \gamma_k^2 = \frac{1+|k|}{M-|k|} \tag{1.192}$$

and

$$\begin{aligned}
&\bar{r}^*(x) = \bar{r}(x, h_k^*(x)) = \sigma^2_{y^{(k)}}(x, h_k^*(x))(1+\gamma_k^2),\\
&\gamma_k = |\bar{m}_e^{(k)}(x, h_k^*(x))|/\sigma^2_{y^{(k)}}(x, h_k^*(x)),\\
&\gamma_k = \sqrt{\frac{1+|k|}{M-|k|}}
\end{aligned} \tag{1.193}$$

Here the parameter γ_k is introduced as a ratio of the upper bound bias to the standard deviation of the derivative estimate for the corresponding ideal scale $h_k^*(x)$. The calculation shows that this parameter is a constant depending only on $M = \min\{m+1, |r|\}$, i.e. it depends only on the power LPA, the smoothness of the function defined by the value of r and the order k of the derivative estimated.

It can be seen also that

$$|\bar{m}_e^{(k)}(x, h(x))| = \begin{cases} < \gamma_k \cdot \sigma_{y^{(k)}}(x,h) \text{ if } h < h_k^*(x), \\ > \gamma_k \cdot \sigma_{y^{(k)}}(x,h) \text{ if } h > h_k^*(x). \end{cases}. \tag{1.194}$$

Latter this inequality is used in order to select the scale close to ideal for the derivative estimation.

Chapter 2

ICI ADAPTIVE VARYING SCALE

2.1 Introduction

Adaptation is now commonly considered as a crucial element of curve estimation procedures, in particular in nonparametric statistical methods. The adaptation used in this chapter is based on the direct nonparametric pointwise image estimation without any preliminary edge recovering. A linear estimator is applied, which is derived from the *LPA* of the image in a window selected by a data-driven way. The non-linearity of the method is incorporated by an adaptive choice of an estimation window allowing to get a near optimal quality of image and edge recovering. The idea of the used adaptation method is simple. The algorithm searches for a largest local vicinity of the point of estimation where the *LPA* assumption fits well to the data. The estimates of the image are calculated for a grid of window sizes and compared. The adaptive window size is defined as the largest of those windows in the grid, which the estimate does not differ significantly from the estimators corresponding to the smaller window sizes. This idea is common for Lepski's approach on the whole. However, the statistics used for comparison of the estimates of different window sizes are defined differently by different authors.

The problem of pointwise adaptive estimation has received a new powerful impetus in connection with a number of new methods developed for adaptive window size selection. Various developments of this idea and various statistical rules are a subject of a thorough study in the papers [74], [81], [97], [114]. It was shown that roughly speaking these new methods are equivalent or nearly equivalent in an asymptotic behavior.

Further this idea have been applied to image estimation [104], [105]. It is assumed that the image is composed of a finite number of regions and the image value is constant within each region. The number of regions, the difference between values of the image function for different regions and the regularity of edges are unknown and may be different for different parts of the image.

This chapter presents a systematic development of the ICI rule for image denoising including basic ideas, algorithms, theoretical performance analysis and simulation results. These algorithms are able to produce a piece-wise smooth surface with a small number of discontinuities in the intensity function or its derivatives. This allows certain desirable features of images such as jumps or instantaneous slope changes to be preserved.

Mainly, we follow the approach and algorithms originally proposed in [37], [38] and further developed in [60], [61], [62], [63]. We name this algorithm the intersection of confidence interval (ICI) rule. The algorithm allows a transparent motivation as well as a simple implementation.

2.2 Idea of ICI

The idea of the algorithm is universal and valid for estimation both the function and the derivative. We explain this idea as applied to estimation of the function and further give details concerning derivative estimation.

The estimation error of the LPA estimator (1.25) can be represented in the form

$$|e(x,h)| = |y(x) - \hat{y}_h(x)| \leq |\bar{m}_e(x,h)| + \left|e^0(x,h)\right|, \tag{2.1}$$

where $|\bar{m}_e(x,h)|$ is the upper bound of the estimation bias (1.182) and $e^0(x,h)$ is a random error with the Gaussian probability density $N(0, \sigma_y^2(x,h))$.

Then

$$\left|e^0(x,h)\right| \leq \chi_{1-\alpha/2} \cdot \sigma_y(x,h)$$

holds with the probability $p = 1-\alpha$, where $\chi_{1-\alpha/2}$ is $(1-\alpha/2)-th$ quantile of the standard Gaussian distribution, and with the same probability

$$|e(x,h)| \leq |\bar{m}_e(x,h)| + \chi_{1-\alpha/2}\sigma_y(x,h). \tag{2.2}$$

It follows from (1.189) that for $h \leq h^*(x)$ the inequality (2.2) can be weakened to

$$|e(x,h)| \leq \Gamma \cdot \sigma_y(x,h), \;\; h \leq h^*(x), \tag{2.3}$$

$$\Gamma = \gamma + \chi_{1-\alpha/2}, \tag{2.4}$$

where

$$\gamma = \frac{1}{M} = \frac{1}{\min\{m+1, r\}}.$$

Now let us introduce a finite set of window sizes:

$$H = \{h_1 < h_2 < .... < h_J\},$$

starting with a quite small h_1, and, according to (2.3), determine a sequence of the confidence intervals $D(j)$ of the biased estimates obtained with the windows $h = h_j$ as follows

$$D(j) = [\hat{y}_{h_j}(x) - \Gamma \cdot \sigma_y(x,h_j), \; \hat{y}_{h_j}(x) + \Gamma \cdot \sigma_y(x,h_j)], \tag{2.5}$$

where Γ is a threshold of the confidence interval.

Then for $h = h_j$ (2.3) is of the form

$$y(x) \in D(j), \tag{2.6}$$

and we can conclude from (2.2) and (2.3) that while $h_j < h^*(x)$ holds for $h = h_j$, $1 \leq j \leq i$, all of the intervals $D(j)$, $1 \leq j \leq i$, have a point in common, namely, $y(x)$.

In the opposite case, when the intersection of the confidence intervals is empty it indicates that $h_j > h^*(x)$. Thus, the intersection of the confidence intervals can be used in order to verify the inequality (1.189).

The following is the ICI statistic, which is used in order to test the very existence of this common point and in order to obtain the adaptive scale value [37], [38]:

Consider the intersection of the intervals $D(j)$, $1 \leq j \leq i$, *with increasing* i, *and let* i^+ *be the largest of those* i *for which the intervals* $D(j)$, $1 \leq j \leq i$, *have a point in common. This* i^+ *defines the adaptive scale and the adaptive* LPA *estimate as follows*

$$\hat{y}^+(x) = \hat{y}_{h^+(x)}(x),\ h^+(x) = h_{i^+}. \tag{2.7}$$

The theoretical analysis produced in [37], [38] shows that the adaptive scale estimate $\hat{y}^+(x)$ gives the best possible pointwise MSE.

In this way it shows that the adaptive window sizes $h^+(x)$ are close to the ideal $h^*(x)$. Simulation experiments confirms this statement.

2.3 *ICI* algorithm

The following algorithm implements the procedure (2.7). Determine the sequence of the upper and lower bounds of the confidence intervals $D(i)$ as follows

$$\begin{aligned} &D(i) = [L_i,\ U_i], \qquad (2.8)\\ &U_i = \hat{y}_{h_i}(x) + \Gamma \cdot \sigma_y(x, h_i),\\ &L_i = \hat{y}_{h_i}(x) - \Gamma \cdot \sigma_y(x, h_i), \end{aligned}$$

where Γ is given by (2.4).

Let

$$\begin{aligned} &\bar{L}_{i+1} = \max[\bar{L}_i,\ L_{i+1}],\ \underline{U}_{i+1} = \min[\underline{U}_i,\ U_{i+1}], \qquad (2.9)\\ &i = 1, 2, ..., J,\ \bar{L}_1 = L_1,\ \underline{U}_1 = U_1 \end{aligned}$$

then the adaptive window length h_i^+ is the largest i when

$$\bar{L}_i \leq \underline{U}_i \qquad (2.10)$$

is still satisfied. This i^+ is the largest of those i for which the confidence intervals $D(i)$ have a point in common as it is discussed above.

We wish to emphasize that this scale *ICI* selection procedure requires a knowledge of the estimate and its variance only.

The *ICI* rule is graphically illustrated in Figure 2.1, where the vertical lines with arrows show the successive intersections of the confidence intervals (1,2), (1,2,3) and (1,2,3,4). Assuming that the intersection with the forth confidence interval (corresponding $h = h_4$) is empty, we obtain the "optimal" adaptive scale $h^+ = h_3$.

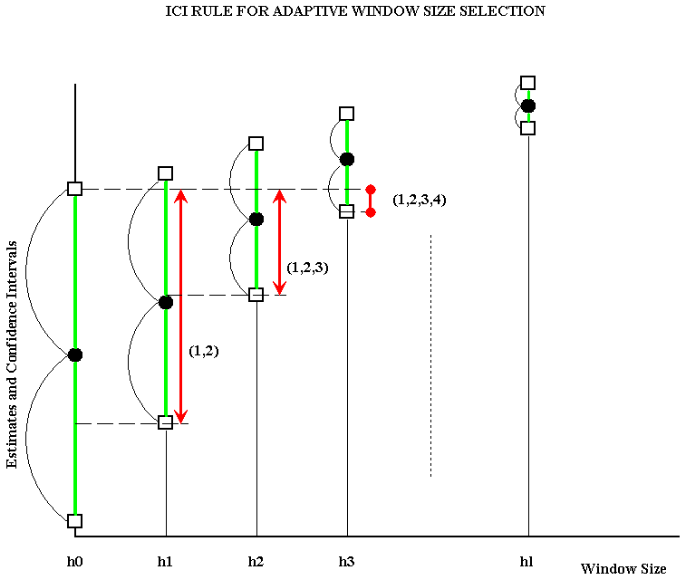

ICI RULE FOR ADAPTIVE WINDOW SIZE SELECTION
Estimates and Confidence Intervals
(1,2)
(1,2,3)
(1,2,3,4)
h0
h1
h2
h3
hl
Window Size

Figure 2.1: Graphical illustration of ICI rule.

2.4 Threshold adjustment

The threshold parameter Γ in (2.8) plays a crucial role in the performance of the algorithm. Too large or too small Γ results in oversmoothing and undersmoothing data.

Let us present some figures for Γ obtained from the theoretical analysis. Assuming $\alpha = 0.05$ or 0.01, then $\chi_{1-\alpha/2} = 2$ or 3 respectively, and we have from (2.4):

$$\Gamma = \begin{cases} 3.0, & \text{for } p = 0.05, \\ 4.0, & \text{for } p = 0.01, \end{cases} \quad \text{for } m = 0, \tag{2.11}$$

$$\Gamma = \begin{cases} 2.7, & \text{for } p = 0.05, \\ 3.7, & \text{for } p = 0.01. \end{cases} \quad \text{for } m = 1.$$

How far can we trust to these figures ? Remind that the formula for γ in (2.4) is derived for the asymptotic as $\Delta \rightarrow 0$. It is assumed also in the analysis that the intensity $y(x)$ is smooth enough in the neighborhood of x, i.e. the first and second order derivatives exist when the zero power ($m = 0$) or the first power ($m = 1$) LPA is applied. In practice, the smoothness of $y(x)$ can not be guaranteed for every x and Δ can be not small enough. Another ambiguity concerns a value of the confidence level α specified for γ calculation. Of course, it changes the corresponding quantile $\chi_{1-\alpha/2}$. Thus, we may conclude, that there are ambiguities, which influence on a calculated threshold value and these ambiguities cannot be resolved in terms of the theoretical analysis only. However, the threshold Γ is a natural design parameter of the algorithm, which can be used in order to refine the algorithm and to adjust it to available observations.

We produced a number of Monte-Carlo simulation experiments in order to verify a role of Γ. In particular, the following observation is interesting. Let the MSE of denoising be minimized on Γ in every Monte-Carlo simulation run. The optimal values of Γ found in this way are of course random. However, these random optimal Γ have very small variations. It means that these optimal random values of Γ depend on statistical properties of the noise and image but not on particular samples occurred in the Monte-Carlo runs. Thus, the optimal Γ is quite robust with respect to random noise components of observations.

The cross-validation (CV) is one of the popular tool developed in quality-of-fit statistics for model selection and adjustment (e.g. [53], [122]). For the linear estimator in the form (1.25) the CV loss function can be represented as a weighted sum of squared residuals (e.g. [60]):

$$I_{CV} = \sum_s \left(\frac{z_s - \hat{y}_{h^+(x_s)}(x_s)}{1 - g_{h^+(x_s)}(x_s, x_s)} \right)^2 . \tag{2.12}$$

For the homogeneous estimator (1.29) $g_{h^+(x_s)}(x_s, x_s)$ is replaced by $g_{h^+(x_s)}(0)$.

This CV criteria can applied for adjustment of Γ. The procedure (2.8)-(2.10) is assumed to be repeated for every $\Gamma \in G$, $G = \{\Gamma_1, \Gamma_2, ...\Gamma_{N_G}\}$, and

$$\hat{\Gamma} = \arg \min_{\Gamma \in G} I_{CV} \tag{2.13}$$

gives the CV adjusted threshold parameter Γ.

The cross-validation in the form (2.12) presents quite a reasonable and efficient selector for Γ. Our attempts to use instead of the cross-validation another quality-of-fit statistics, in particular the C_P, Akaike criteria and its modifications (see e.g. [53]), which are different from I_{CV} only by the used weights of the residuals, have not shown an improvement in the estimation accuracy.

The adjusted adaptive LPA estimation consists of the following basic steps:

1. Set $\Gamma = \Gamma_l$, $l = 1, 2, ..., N_G$ and $x = x_s$, $s = 1, 2, .., n$.

2. For $h = h_i$, $i = 1, ..., J$, calculate the estimates $\hat{y}_h(x_s)$, the adaptive scale $h^+(x_s)$ and the estimate $\hat{y}_{h^+(x_s)}(x_s))$.

4. Repeat Step 2 for all x_s, $s = 1, 2, .., n$, and Γ_l, $l = 1, 2, ..., N_G$.

5. Find $\hat{\Gamma}$ from (2.13) and select estimates $\hat{y}(x_s, h^+(x_s)$ corresponding to $\hat{\Gamma}$ as the final ones.

The standard deviation of the noise σ used in the formula for $\sigma(x, h)$ in (2.8) should be estimated independently from the ICI algorithm.

In particular, the following well-known estimator proves to be efficient

$$\hat{\sigma} = \{\text{median}(|z_s - z_{s-1}| : s = 2, .., n)\}/(\sqrt{2} \cdot 0.6745). \tag{2.14}$$

The average $\frac{1}{N-1}\sum_{n=2}^{N}(z_s - z_{s-1})^2$ could also be applied as an estimate of σ^2. However, we prefer a median (2.14) as a robust estimate.

2.5 Adaptive varying windows for derivative estimation

All speculations and ideas used for derivation of the above ICI algorithm are valid for estimation of the derivative provided that $\hat{y}_h(x)$ and $\sigma_y(x,h)$ are replaced by $\hat{y}_h^{(k)}(x)$ and $\sigma^2_{y^{(k)}}(x,h)$, respectively. Thus, the confidence intervals used in the ICI algorithm (2.8) are

$$\begin{aligned} &D(i) = [L_i,\ U_i], \\ &U_i = \hat{y}_{h_i}^{(k)}(x) + \Gamma_k \cdot \sigma_{y^{(k)}}(x,h_i), \\ &L_i = \hat{y}_{h_i}^{(k)}(x) - \Gamma \cdot \sigma^2_{y^{(k)}}(x,h_i), \end{aligned} \tag{2.15}$$

where accordingly to (2.4), (1.194) and (1.193)

$$\Gamma_k = \gamma_k + \chi_{1-\alpha/2} = \sqrt{\frac{1+|k|}{M-|k|}} + \chi_{1-\alpha/2}$$

is a threshold of the confidence interval.

It is interesting to give figures for Γ_k, similar to presented in (2.11). Let $k = (1,0)$ or $k = (0,1)$, $|k| = 1$, i.e. the first derivative on X or on Y is estimated and $|k| \leq m \leq r-1$. Then, $\Gamma_k = \sqrt{\frac{1+|k|}{m+1-|k|}} + \chi_{1-\alpha/2}$ and we obtain:

$$\begin{aligned} &\Gamma_1 = \begin{cases} 3.4, & \text{for } p = 0.05, \\ 4.4, & \text{for } p = 0.01, \end{cases} \quad \text{for } m = 1, \\ &\Gamma_1 = \begin{cases} 3.0, & \text{for } p = 0.05, \\ 4.0, & \text{for } p = 0.01. \end{cases} \quad \text{for } m = 2. \end{aligned} \tag{2.16}$$

Thus comparison with (2.11) shows that the derivative estimation requires larger values of Γ.

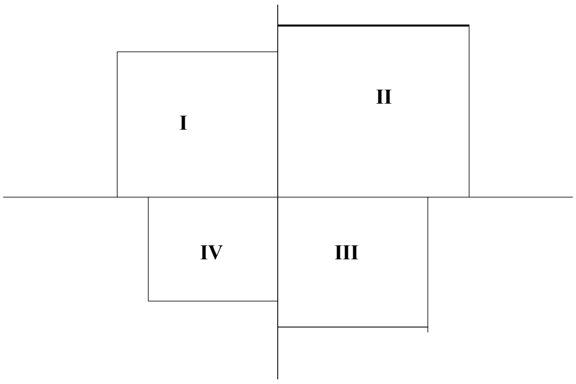


Figure 2.2: Four quadrant windows I,II,III and IV used for directional window size selection by the ICI rule.

2.6 Multiple window estimation

Different ideas can be used for a selection of window function $w_h(x)$ for processing of $2D$ image signals. The simplest and standard one assumes that a symmetric square window $w_h(x)$ is applied for every pixel and the size of the window (parameter h) is the only varying parameter to be found.

A more complex approach assumes that the varying window is composed from a number of separate segments, say from four quadrants shown in Figure 2.2. The centre of the window is the initial point of the Cartesian coordinate system $(0, 0)$. Each segment is a square covering a part of the corresponding quadrant. It is assumed that this initial point $(0, 0)$ is the centre of the LPA estimate for each square segment. The sizes of these squares are independent parameters of the combined window.

Two stage estimation is considered. On the first stage, the ICI rule is used for independent selection of the sizes for these separate four windows. On the second stage, the independently found estimates are combined in the final one.

There are a number of ways how to fuse estimates obtained for the separate window segments into the final estimate.

Some of our simulation results presented in this chapter are obtained for the following final estimate:

$$\hat{y}(x) = \sum_{j=[I,II,III,IV]} \lambda_j \hat{y}_{j,h_j^+(x)}(x), \tag{2.17}$$

$$\lambda_j = \frac{std_j^{-2}(x)}{std^{-2}(x)},\ std^2(x) = \sum_{j=[I,II,III,IV]} std_j^2(x),$$

where $\hat{y}_{j,h_j^+(x)}(x)$ are the quadrant estimates with the ICI rule adaptive scale $h_j^+(x)$. These estimates are obtained respectively for the windows I, II, III, IV in Figure 2.2. Further λ_j and std_j are the weights and the standard deviations of these estimates $\hat{y}_{j,h_j^+(x)}(x)$. In the estimate (2.17) we use a linear fusing of the estimates with the inverse variances of the estimates as weights. This rule can be derived from the maximum likelihood method assuming that the estimates to fuse are Gaussian, unbiased and independent.

Similar multiple window estimates have been applied in [37], [38] and [60] for $1D$ function estimation.

We may fuse the quadrant estimates in many other ways. In particular, as:

(a) Ordinary mean

$$\hat{y}(x) = \frac{1}{4}\sum_j \hat{y}_{j,h_j^+(x)}(x), \tag{2.18}$$

(b) Ordinary median

$$\hat{y}(x) = median_j\ \{\hat{y}_{j,h_j^+(x)}(x)\}, \tag{2.19}$$

(c) Weighted mean

$$\hat{y}(x) = \frac{1}{\sum_j [h_j^+(x)]^2} \sum_j [h_j^+(x)]^2 \cdot \hat{y}_{j,h_j^+(x)}(x). \tag{2.20}$$

It used in the last estimate here that the estimate variance is inverse proportional to the squared scale parameter h, $std^2 \sim 1/h^2$.

2.7 Combined-window estimates

In multiple window estimation we calculate the estimate for different windows and fuse these estimates together in the final one. In this section we consider a different idea.

The estimation also includes two successive stages. On the first stage the simple mean estimate with ICI rule is used for the varying scale selection for the four quadrant estimates, i.e. for the quadrants Q_I,Q_{II},Q_{III},Q_{IV} as they are shown in Figure 2.2. These adaptive size quadrants compose a support for the second stage estimate, which is defined as

$$Q = Q_I \cup Q_{II} \cup Q_{III} \cup Q_{IV}. \tag{2.21}$$

As the ICI is a point-wise rule the supports Q_j are different for different x. Thus, the combined support Q is also pixel-wise varying and can have a very unusual form.

On the second stage the final estimate is obtained by applying any estimator with the found varying combined support $Q(x)$. This second stage estimate is a final one. Examples of this two stage adaptive estimator are demonstrated in [68], where the orthogonal transforms (wavelet and DCT) are applied as second stage estimators . The second stage median filters are reported in [61], [69].

In the combined window estimation we use the mean ICI adaptive estimator as an auxiliary tool only in order to form $Q(x)$. The mean estimator can be combined with the ICI rule in a very efficient way and results in fast algorithms. The results shown in the cited papers confirm the efficiency this type of the algorithms.

Chapter 3

DENOISING EXPERIMENTS

The LPA estimators and the ICI rule for scale selection introduced above define a conceptual framework, basic ideas of adaptive algorithms developed for noise reduction. A number of implementations and modifications of these algorithms have been developed and studied.

In this chapter we follow the basic framework of the LPA estimates as they have been presented above.

The major parameters of these estimates are as follows:

(a) The threshold Γ;

(b) The power m of the LPA;

(c) The set H defining a grid of the window sizes.

In image perception the image edges are of importance and also the edges are important on their own as a tool for pattern recognition, classification, etc. The derivatives are basic tools for many edge detection algorithms. We show here the use of the ICI adaptive LPA for derivative estimation.

It is demonstrated also that the adaptive ICI varying windows carry important information concerning smoothness, curvature and singularities of the image intensity function. These adaptive varying window sizes can be used directly for edge detection.

In this chapter we present simulation results, which illustrate the efficiency of the ICI rule for different problems and give an insight into an influence of the major algorithm parameters.

When the power m of LPA is equal or higher than one then the LPA estimates of the derivatives are available. We consider two versions of the adaptive scale derivative estimators. In the first one, the ICI adaptive varying window sizes found for the function estimation are used for the derivative estimation. We call this method the *function-ICI* adaptive derivative estimation. In the second one, the ICI rule is applied directly to the derivative estimates in order to find the adaptive window sizes specifically for the derivative estimation. It makes a big deal of difference with the *function-ICI* adaptive estimates. We call this second method the *derivative-ICI* adaptive derivative estimation.

Here we use the function and derivative estimators in the form of the $2D$ convolution (1.44) with the square rectangular window function w. It is assumed that $\Delta_1 = \Delta_2 = 1$.

We differ five types of the square windows w: four nonsymmetric ones as presented in Figure 2.2 and the symmetric square one with the origin coordinate point as a center of this square. We use the following notation for these supports Q_j, $j = 0, 1, ..., 4$, where Q_0 corresponds to the symmetric window and Q_j with $j = 1, ..., 4$ respectively to the quadrants I, II, III, IV.

We present the simulation for the uniform and the Gaussian windows w, which are specified by the formulas:

$$w(x) = \begin{cases} 1,\ |x_1| \leq 1,\ |x_2| \leq 1,\ Q_0, \\ 1,\ -1 < x_1 \leq 0,\ 0 \leq x_2 < 1,\ Q_1, \\ 1,\ 0 \leq x_1 < 1,\ 0 \leq x_2 < 1,\ Q_2, \\ 1,\ 0 \leq x_1 < 1,\ 0 \leq x_2 < -1,\ Q_3, \\ 1,\ -1 < x_1 \leq 0,\ -1 < x_2 \leq 0,\ Q_4, \end{cases} \tag{3.1}$$

and

$$w(x) = \begin{cases} \exp(-||x||^2/2\sigma_w^2),\ |x_1| \leq 1,\ |x_2| \leq 1,\ Q_0, \\ \exp(-||x||^2/2\sigma_w^2),\ -1 < x_1 \leq 0,\ 0 \leq x_2 < 1,\ Q_1, \\ \exp(-||x||^2/2\sigma_w^2),\ 0 \leq x_1 < 1,\ 0 \leq x_2 < 1,\ Q_2, \\ \exp(-||x||^2/2\sigma_w^2),\ 0 \leq x_1 < 1,\ 0 \leq x_2 < -1,\ Q_3, \\ \exp(-||x||^2/2\sigma_w^2),\ 1,\ -1 < x_1 \leq 0,\ -1 < x_2 \leq 0,\ Q_4. \end{cases} \tag{3.2}$$

It follows from (3.1) and (3.2) that for any integer $h > 0$ the weights $w(x/h)$ have $(2h-1)^2$ and h^2 nodes for symmetric and nonsymmetric windows respectively.

The scale set H is defined by the equation

$$H = \{m+1,\ \lceil 1.45.^{[m+1:9]} \rceil\}, \tag{3.3}$$

where $\lceil x \rceil$ rounds x to the nearest integers towards infinity. We start H from $m+1$ in order to have in H a number of items sufficient for the polynomial approximation of the power m. In particular for $m = 0$ it gives

$$H = \{1,\ 2,\ 3,\ 4,\ 5,\ 7,\ 10,\ 14,\ 20,\ 29\}. \tag{3.4}$$

The set H (3.3) is sparse towards large h and quite dense for small h. In this way we obtain H of quite a small cardinal number, what enables one to produce fast calculations. A high density of H toward small values of h makes possible a good resolution of small image details.

The multiple window estimates used as final ones are defined by the average (2.17), where the estimate of the symmetric window is also included.

It is assumed that the additive Gaussian noise has the standard deviation $\sigma = 0.1$ and the image intensity is normalized to the maximum value equal to 1.

3.1 Filtering

We present simulation results, which illustrate the efficiency of the ICI rule and give an insight into behavior of the estimates of the image intensify as well as the adaptive varying window sizes. Experiments are performed on two test images: "Square" (8 bit binary 128×128 image) and "Cameraman" (8 bit gray-scale 256×256 image).

3.1.1 ICI-varying window sizes, $m = 0$, "Square"

Let us start from a simple binary image "Square", which is composed from a central square of the size 64×64 with the intensity equal to zero and a background equal to one beyond the central square. Images of the true and noisy "Square" are shown in Figure 3.1. The zero intensity (black) smaller square is centered on the large background (white) square.

Denoising results are shown in Figures 3.2-3.9. The first two figures, Figure 3.2 and Figure 3.3, are obtained by the $ICI - LPA$ estimates of the zero power $m = 0$ and the threshold $\Gamma = 3.5$. Figure 3.2 shows the symmetric window estimate and the quadrant estimates marked by Q_j, $j = 0, 1, ..., 4$, respectively. The final combined estimate is defined by the formula (2.17), where the symmetric window estimate is also included.

The figures of the intensity estimates are equipped with values of the accuracy criteria. Here $RMSE$ means the root mean squared error,

$$RMSE = \sqrt{\frac{1}{n}\sum_s (y(x_s) - \hat{y}_{h^+(x_s)}(x_s))^2}, \tag{3.5}$$

and $ISNR$ means the improvement in SNR in dB, i.e.

$$ISNR = 20\log_{10}(\hat{\sigma}/RMSE), \tag{3.6}$$

$\hat{\sigma}^2 = \frac{1}{n}\sum_s (z(x_s) - y(x_s))^2$ is the estimate of the variance of the additive random noise. These criteria are calculated for the symmetric and all quadrant window estimates.

Figure 3.3 presents the varying adaptive window sizes produced by the ICI rule for each of the all five (symmetric and nonsymmetric windows) estimates and used in the estimates shown in Figure 3.2.

For clarity we complete Figures 3.2 and 3.3 by cross-sections made through the centre of "Square" in direction parallel to the horizontal axis X. These cross-sections are given for the estimates as well as for the adaptive window sizes in Figures 3.4 and 3.5, respectively. The last figure for the adaptive window sizes demonstrates that the ICI adaptive window sizes are very sensitive with respect to discontinuities in the image. Even more, this ICI sensitivity is directional. It results from fact that increasing of h means that the quadrant window is growing in some particular direction. Thus h increasing means the window growing in North-West for the quadrant for Q_1, North-East for the quadrant Q_2, South-East for the quadrant Q_3, and South-West for the quadrant for Q_4.

For an accurate interpretation of Figure 3.5 we note that the true image cross-section inside and outside of the image support is given by the expression

$$y(64,k) = \begin{cases} \ \ 1, \ \text{for } 1 \leq k \leq 32, \ 97 \leq k \leq 128, \\ 0, \ \text{for } 33 \leq k \leq \ 96, \ k < 0, \ k > 128. \end{cases}$$

Remind that the LPA estimate with $m = 0$ is a sample mean. Let us look at the curves in Figures 3.4 and 3.5 corresponding the estimate in the quadrant Q_1.

We consider the estimates for the argument value k, $1 \leq k \leq 128$. For $1 \leq k \leq 32$ the true function is equal to one and using the data from Q_1 for estimation of this constant value means that data from the left hand-side of k can be used only. The scale of data available for this estimation is equal to k, i.e. the ideal scale is the linear growing function of k. We can see a stepwise approximation of this ideal linear function in Figure 3.5 as it is found by the ICI rule. This stepwise approximation as well as the restriction on the used maximum value of the scale are defined by the window sizes given in H.

As soon as $k > 32$ the true function becomes equal to zero and continuous to be zero for $33 \leq k \leq 96$. For estimation of this constant using the ideal window in Q_1 we use only observations for the points equal or less than k, provided that $33 \leq k \leq 96$. Thus, again we obtain the ideal window size as a linear growing function of k starting from small values. This situation is repeated for the third interval $97 \leq k \leq 128$ also. It is seen in Figure 3.5 that the ICI rule gives the adaptive window sizes, which perfectly approximate this ideal piece-wise linear growing window sizes.

The same interpretation is valid for the ICI adaptive varying window sizes shown in Figure 3.5 for the window Q_4, while the opposite behavior of the varying window having decreasing sizes are valid for the windows Q_2 and Q_3.

A shape of the ideal as well as of adaptive window size is quite different for the symmetric window estimate (see corresponding curves in Figure

3.5). The maximum peaks of the window sizes in the middle of the three subintervals is a specific feature of the symmetric window estimate.

The ICI rule is able to select smaller windows in a neighborhood of the intensity jumps and larger window for areas of flat image intensity. However, the different quadrants give larger or smaller values of the window size for the same jump point depending on what side of the jump neighborhood is used for estimation. As a result the adaptive window sizes shown in Figure 3.5 delineate the intensity nearly perfectly in a complete agreement with as it could be done provided that the intensity function is known in advance.

The cross-sections of the estimates shown in Figure 3.4 demonstrate the efficiency of the ICI adaptive window size estimates in neighborhood of the jumps. The nonsymmetric window estimates caught the jumps perfectly from the left and right sides provided that the corresponding left-, right-window estimators are applied. Note that the symmetric window size estimates are not able to achieve this sort of perfect behavior near the jump points. The symmetric estimator always tends to oversmooth the discontinuity points. The advantage of these symmetric-window estimates has a place on flat sections of the image, where the average calculated over left and right observation points is able to improve the noise suppression.

In Figure 3.3 small and large window sizes are shown by black and white, respectively. Again we are able to see a difference in performance of the different quadrant estimates. The adaptive window sizes correspond to the intuitively clear behavior of the ideal varying window obtained provided that the true image is known. Thus, the window sizes delineate edges of the true image, and the variations of the window sizes provides a shadowing of the image from different sides of the image in the directions defined by the type of the used nonsymmetric quadrant estimator. The images in Figure 3.3 show also the boundaries of the background. It happens because it is assumed that out of the "Square" image the intensity is equal to zero. In this way, the image intensity used in the estimation outside the image support has a jump at the outside boundaries of the image. The corners and backsides of the boundaries are clearly seen in the images of the adaptive window sizes for the corresponding quadrants.

It was noticed that the ICI adaptive window sizes, in particular for small Γ, can be corrupted by spikes, which are erroneously isolated small values of the adaptive varying window sizes [60]. Figures 3.6 and 3.7 illustrate this effect. They are depicted for the scenario presented in Figures 3.2-3.3 provided that smaller $\Gamma = 2.0$ is used. If we compare images of Figure 3.7 with the corresponding images of Figure 3.3 we can see multiple black spots in Figure 3.7 showing these window size spikes.

However, for many occasions these spikes do not degrade the performance and even more we obtain a better performance for smaller Γ even visually the adaptive window sizes look quite noisy. The comparison of $RMSE$ and $IMSE$ in Figures 3.2 and 3.6 are definitely in favor of the estimate with smaller value of Γ for all estimates. For the final combined estimate we have for $RMSE = 0.0154$ and $IMSE = 16.3$ dB for $\Gamma = 2.0$ versus $RMSE = 0.0243$ and $IMSE = 12.3$ dB for $\Gamma = 3.5$.

Actually, these isolated spikes of the window sizes have different locations in different quadrant estimates and as a result they do not influence the final estimate in a crucial way. In the same time the estimates with smaller Γ are more sensitive with respect to irregularities in images and give better performance near the edges.

Nevertheless, we note that these spikes in the window sizes can be eliminated by increasing the value of Γ. We have seen this effect comparing Figure 3.3 versus Figure 3.7. However, as we sought it does not improve the quality of estimation. The problem of Γ selection is important for the accuracy improvement. In any case an improvement in the appearance of the adaptive window size figures does not guarantee the accuracy improvement.

The preliminary filtering of $h^{+}(x)$ considered as a function of x also can be used as another tool in order to suppress the spikes. Figures 3.8-3.9 present filtering results of data given in Figures 3.7 by the $2D$ median filter 3×3. In part the black sports (spikes of the window size) are taken out in Figure 3.9. However, as it follows from the criteria values shown for the estimates in Figures 3.6 and 3.8, there is a degradation in the accuracy estimation as a result of this prefiltering. It is seen for all quadrant and final estimates. Thus, the prefiltering of the ICI adaptive window sizes is not able to guarantee an improvement in the quality of denoising.

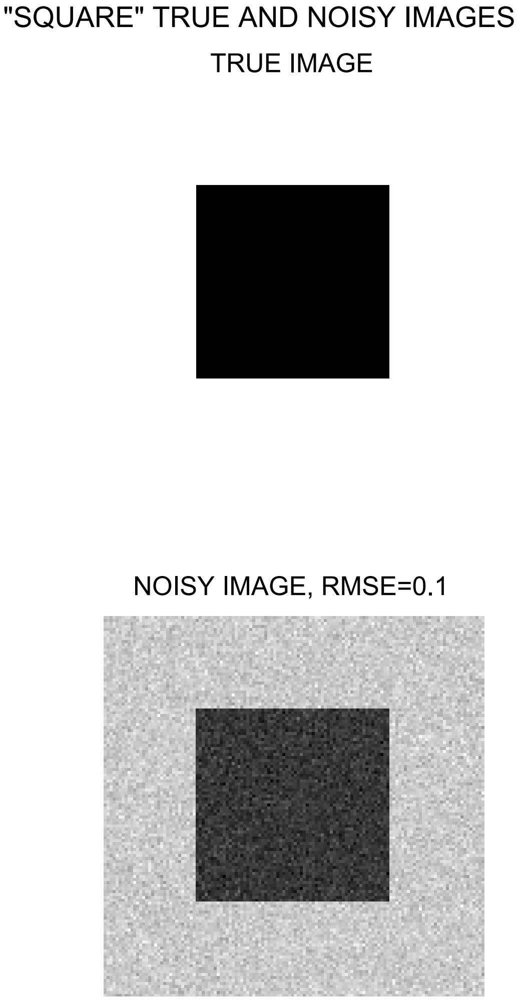


Figure 3.1: True and noisy images of ”Square”

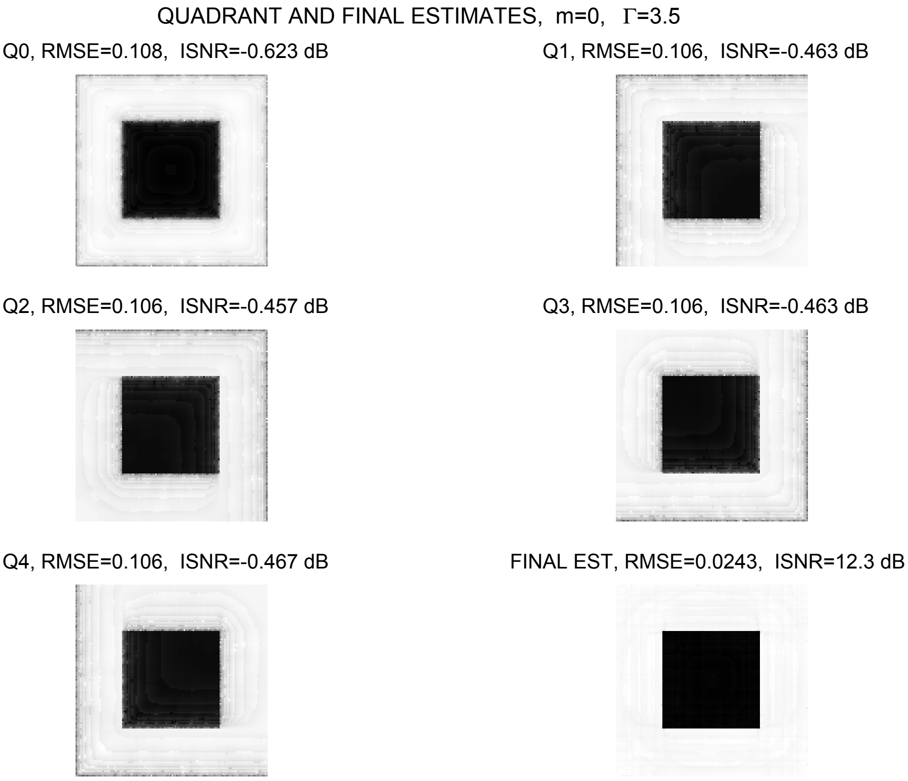


Figure 3.2: "Square" image estimates. The estimates are given for the symmetric window estimate (Q_0), the nonsymmetric quadrant window estimates (Q_1,Q_2,Q_3,Q_4), and for the final estimate formed from the quadrant and symmetric window estimates. $RMSE$ and $ISNR$ specify the accuracy of the corresponding estimates.

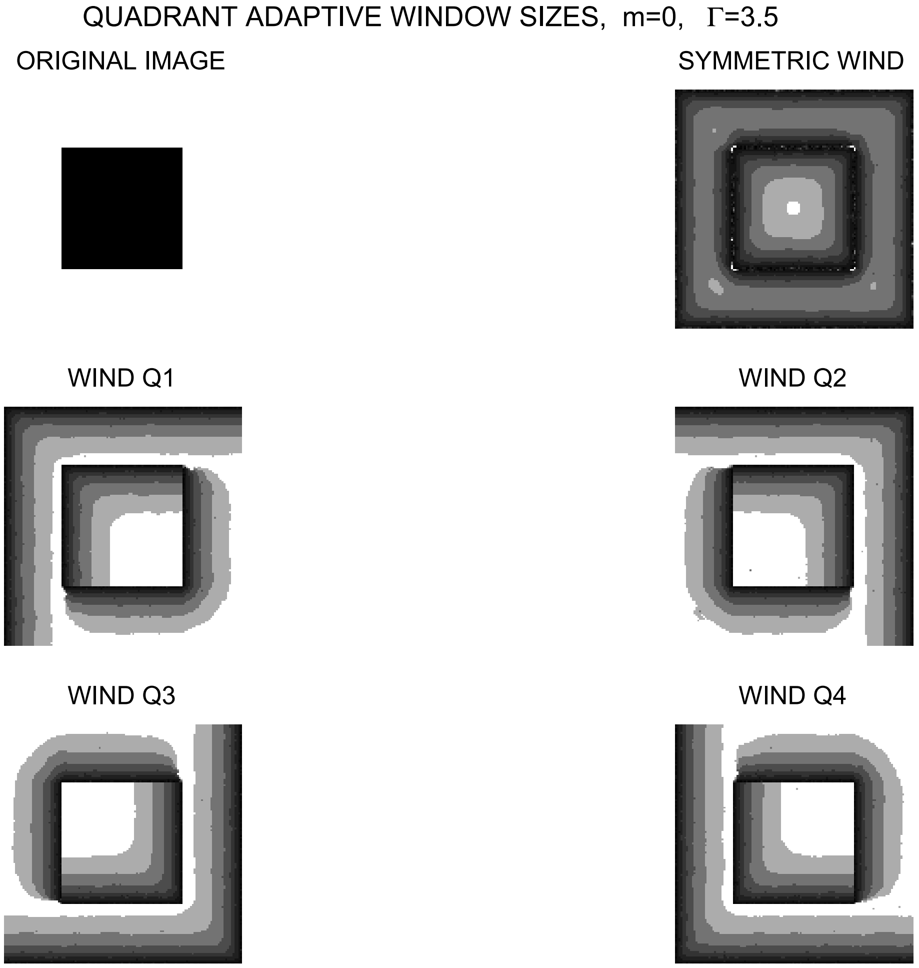


Figure 3.3: "Square" true image and adaptive varying window sizes obtained by the *ICI* rule for the symmetric and quadrant windows estimates.

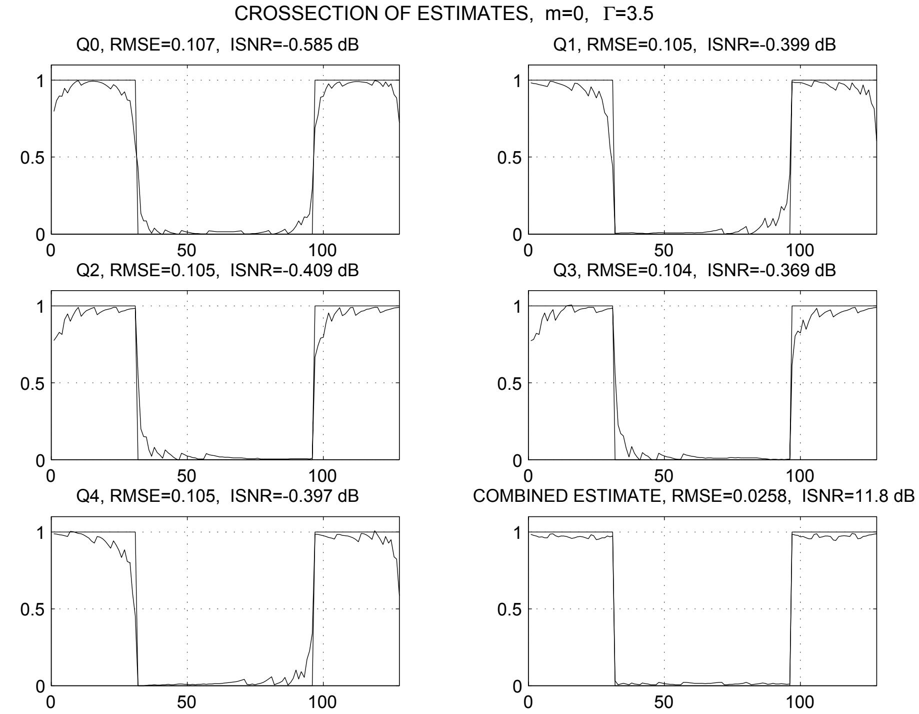


Figure 3.4: Curves show the cross-section of the true "square" image and the estimates. The cross-section are given for symmetric window estimate (Q_0), nonsymmetric quadrant window estimates (Q_1,Q_2,Q_3,Q_4), and for the final estimate formed from the quadrant and symmetric window estimates. $RMSE$ and $ISNR$ specify the accuracy of the corresponding estimates of the $2D$ image.

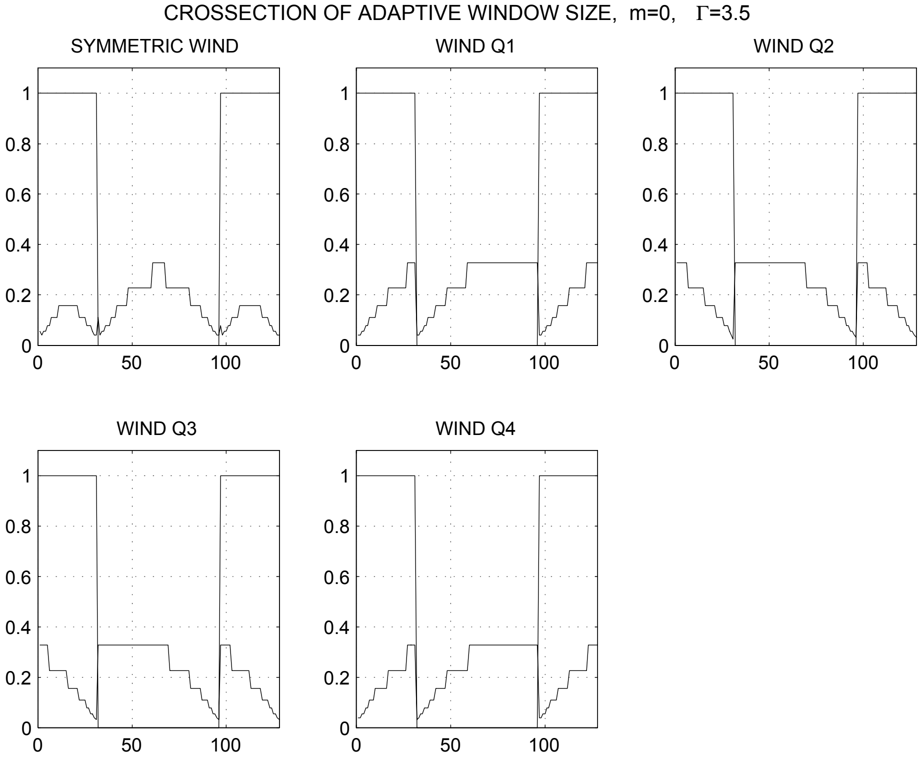


Figure 3.5: Curves show the cross-sections of the true image "square" and the adaptive window sizes as they are produced by the ICI rule. The cross-sections are given for the symmetric window estimate (Q_0), nonsymmetric quadrant window estimates (Q_1,Q_2,Q_3,Q_4). $RMSE$ and $ISNR$ specify the accuracy of the corresponding estimates of the $2D$ image. The adaptive window sizes are normalized to the image size. The curves show the normalized window size $h/128$.

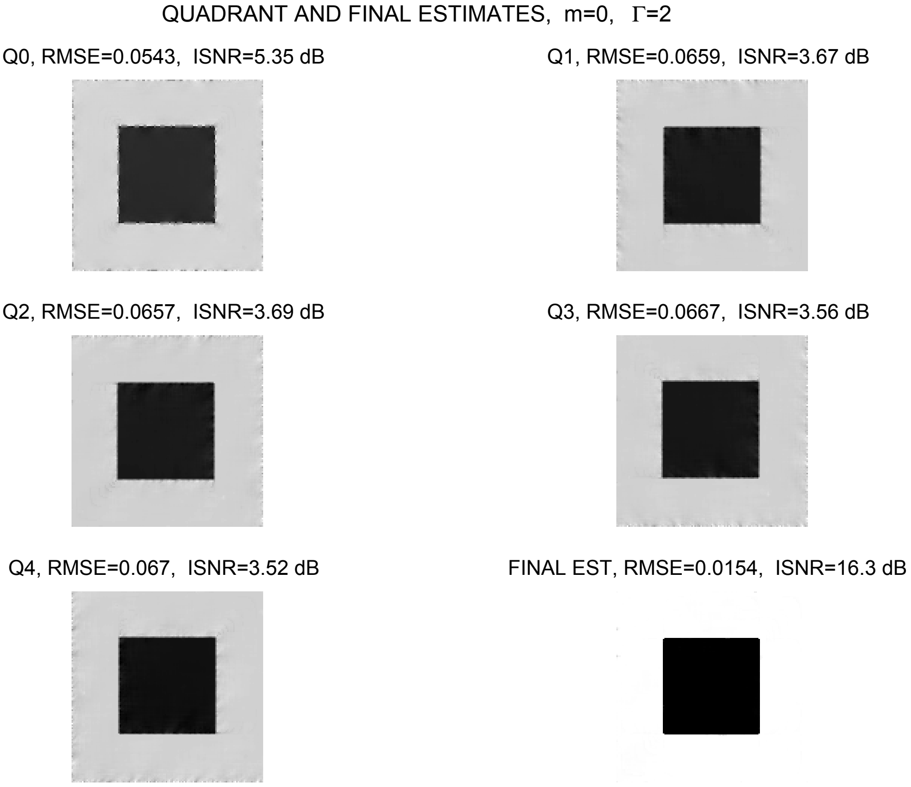

QUADRANT AND FINAL ESTIMATES, m=0, Γ=2
Q0, RMSE=0.0543, ISNR=5.35 dB
Q1, RMSE=0.0659, ISNR=3.67 dB
Q2, RMSE=0.0657, ISNR=3.69 dB
Q3, RMSE=0.0667, ISNR=3.56 dB
Q4, RMSE=0.067, ISNR=3.52 dB
FINAL EST, RMSE=0.0154, ISNR=16.3 dB

Figure 3.6: "Square" images show the image estimates. The estimates are given for the symmetric window estimate (Q_0), the nonsymmetric quadrant estimates (Q_1,Q_2,Q_3,Q_4), and for the final estimate. $RMSE$ and $ISNR$ specify the accuracy of the corresponding estimates.

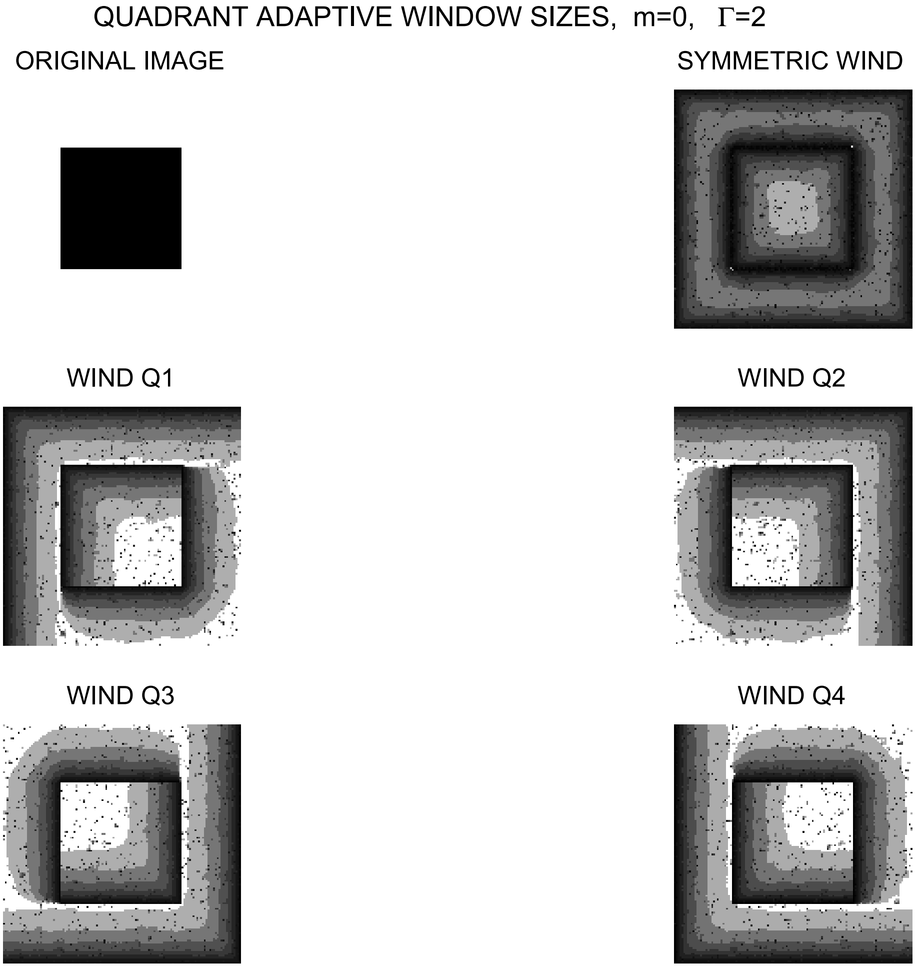


Figure 3.7: "Square" true image and the *ICI* adaptive varying window sizes for the symmetric and the quadrant image estimates.

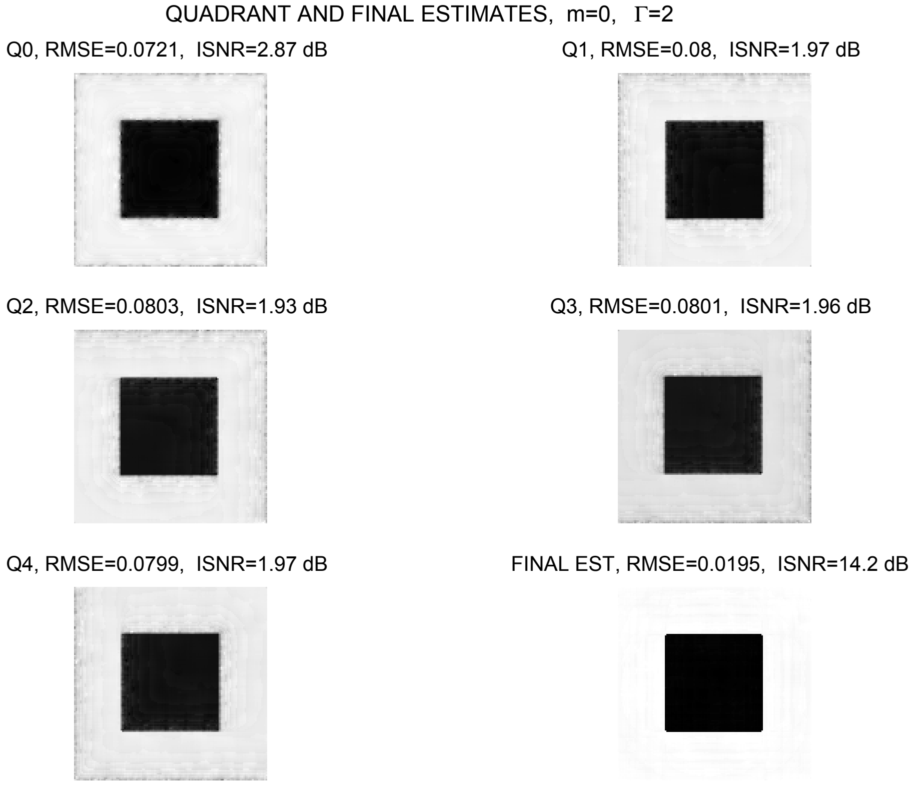

QUADRANT AND FINAL ESTIMATES, m=0, Γ=2
Q0, RMSE=0.0721, ISNR=2.87 dB
Q1, RMSE=0.08, ISNR=1.97 dB
Q2, RMSE=0.0803, ISNR=1.93 dB
Q3, RMSE=0.0801, ISNR=1.96 dB
Q4, RMSE=0.0799, ISNR=1.97 dB
FINAL EST, RMSE=0.0195, ISNR=14.2 dB

Figure 3.8: "Square" image estimates obtained after prefiltering the ICI adaptive window sizes. The estimates are given for: the symmetric window (Q_0), the nonsymmetric quadrant (Q_1,Q_2,Q_3,Q_4), and for the final estimates. The ICI adaptive window sizes are filtered by the $2D$ median filter, 3×3 . $RMSE$ and $ISNR$ specify the accuracy of the corresponding estimates.

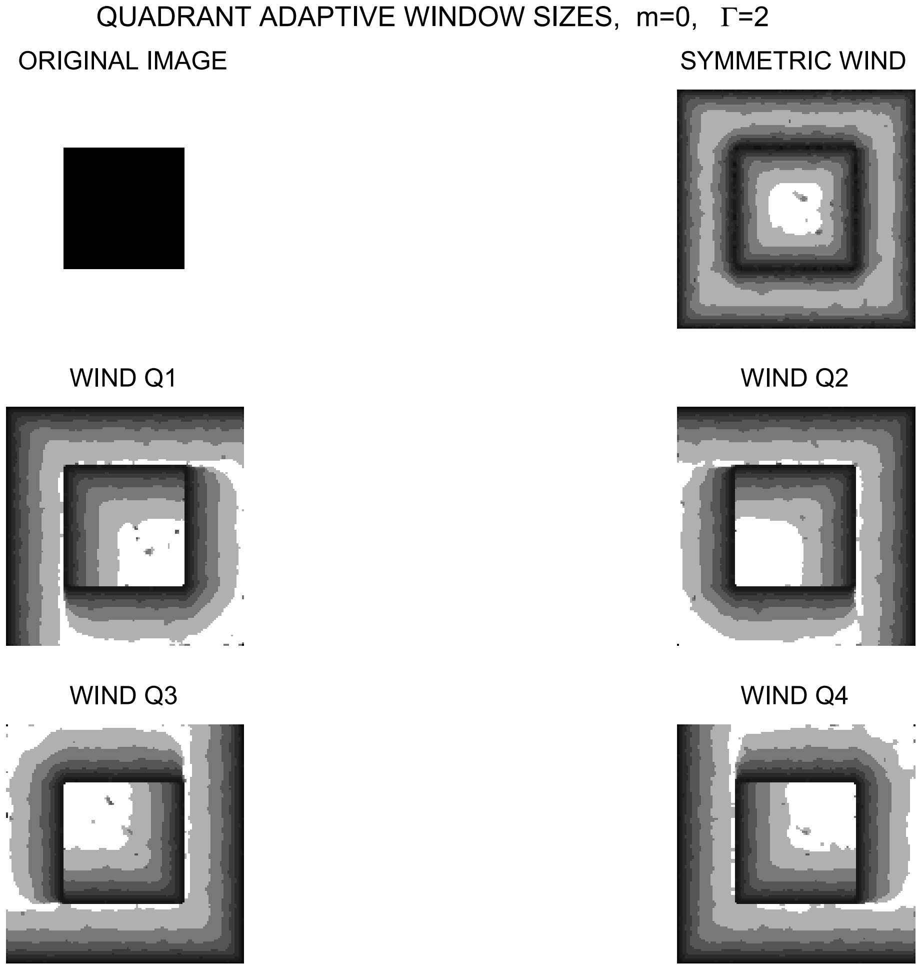


Figure 3.9: "Square" images show the ICI adaptive window sizes after prefiltering. The varying windows are given for the symmetric window estimate (Q$_0$), the nonsymmetric quadrant estimates (Q$_1$,Q$_2$,Q$_3$,Q$_4$). $2D$ median filter 3×3 is used for the prefiltering. $RMSE$ and $ISNR$ specify the accuracy of the corresponding estimates.

3.1.2 ICI-varying window sizes, $m = 0$, "Cameraman"

Let us go further and consider the result for the second test-image "Cameraman". True and noisy "Cameraman" images are shown in Figure 3.10. The simulation experiments are arranged similarly to as it was done for "Square" and shown in Figures 3.11-3.16. The first two Figure 3.11 and Figure 3.12 are obtained for the LPA estimates of the zero power, $m = 0$, and the threshold $\Gamma = 3.5$. Figure 3.11 shows the symmetric window estimate, the quadrant estimates marked by the corresponding Q_j, $j = 0, 1, ..., 4$, and the final combined estimate. The accuracy criteria values are also shown in the figures as it was above. In Figure 3.12 small and large window sizes are shown by black and white, respectively. The ICI based adaptive window sizes are very sensitive with respect to edges in images. The ICI rule selects smaller windows in a neighborhood of the intensity changes and larger window for areas of flat image intensity. Again we can see that the ICI sensitivity is directional and the varying window sizes are different for different quadrant estimates giving a directional shadows to the image edges. The adaptive window sizes shown in Figure 3.12 delineate and shade contours of the cameraman, camera, and other image details. The obtained window sizes actually correspond to the intuitively clear behavior of the varying window size relevant to the smoothing of the data if the true image is known.

Figures 3.13 and 3.14 are similar to Figures 3.11 and 3.12 but obtained for the smaller threshold $\Gamma = 2.0$. We can observe that the ICI adaptive window sizes in Figure 3.14 are corrupted by the spikes shown by multiple black spots, which are erroneously isolate small values of the adaptive window sizes. However, as it is clear from values of $RMSE$ and $ISNR$ given in Figures 3.11 and 3.13 the accuracy achieved for this smaller $\Gamma = 2.0$ is much better than that for $\Gamma = 3.5$. However, the adaptive ICI windows look in much more appealing way in Figure 3.11 ($\Gamma = 3.5$) than in Figure 3.14 ($\Gamma = 2$).

The spike in the ICI adaptive windows $h^{+}(x)$ can be filtered out and calculation of the new estimates obtained for these new window sizes is done. Again we apply here the 3×3 median pre-filter. Figures 3.15 and 3.16 illustrate the corresponding effects. As a result the spikes, at least in part, are eliminated from the window size images. However, the overall quality of image denoising becomes worse as it is clear from the corresponding values of $RMSE$ and $IMSE$ in Figures 3.13 and 3.15 respectively. Thus, the prefiltering of $h^{+}(x)$ is not able to guarantee an improvement of image filtering.

It has been demonstrated that the algorithm performance depends essentially on the threshold Γ. Let us consider optimization of this parameter. The performance of imaging is characterized by a set of criteria:

(1) $RMSE$ defined by (3.5)

$$RMSE = \sqrt{\frac{1}{\#}\sum_{x}(y(x) - \hat{y}_{h^{+}(x)}(x))^{2}}; \tag{3.7}$$

(2) SNR (dB) defined by

$$SNR = 10\log_{10}\frac{\sum_{x}|y(x)|^{2}}{\sum_{x}|y(x) - \hat{y}_{h^{+}(x)}(x)|^{2}}; \tag{3.8}$$

(3) $ISNR$ (dB) defined by (3.6)

$$ISNR = 20\log_{10}(\frac{\hat{\sigma}}{RMSE}); \tag{3.9}$$

(4) Peak signal-to-noise ratio ($PSNR$ in dB)

$$PSNR = 10\text{log}_{10}(\frac{max_{x}y(x)}{RMSE}); \tag{3.10}$$

(5) Mean absolute error (MAE):

$$MAE = \frac{1}{\#}\sum_{x}|y(x) - \hat{y}_{h^{+}(x)}(x)|; \tag{3.11}$$

(6) Maximum absolute difference (error) ($MAX - DIF$)

$$MAX - DIF = \max_{x}|y(x) - \hat{y}_{h^{+}(x)}(x)|. \tag{3.12}$$

These criteria allow to evaluate the performance of the algorithm quantitatively, while $PSNR$ is treated as a criterion linked with a visual image perception. However, the visual evaluation is considered as an independent performance criterion required to be evaluated.

In our study we consider the above six criteria as functions of Γ and use them for optimization of Γ.

Figures 3.17 and 3.18 show these criteria for the test-images "Square" and "Cameraman", respectively, for the simulation conditions presented above, provided that Γ is varying. First of all note, that the final estimates for all criteria and for the both images give best performance. The curves corresponding to the quadrant (Q_1,Q_2, Q_3, Q_4) estimates demonstrate nearly equivalent results and actually are overlapping. Interesting to remark, that for the "Square" image the symmetric window estimate shows worse performance than the quadrant estimates. For the "Cameraman" image the symmetric window estimate is better, than the quadrant estimates. It can be explained by the fact the "Square" image has a discontinues binary intensity, where the nonsymmetric quadrant estimates are able to make a better job.

Concerning optimization of Γ we can see the following. For the "Square" image the optimum values of $PSNR$, $RMSE$, SNR, $ISNR$ for the final estimate are achieved at $\Gamma = 2$, while a bit better value of MAE is at $\Gamma = 1.5$. The optimization of the quadrant and symmetric window estimates is in favor of smaller value of $\Gamma = 1.0$. Overall we accept $\Gamma = 2$ to be the best choice for "Square" image estimation. Thus, Figures 3.6 and 3.7 demonstrate the performance of the algorithm for this best $\Gamma = 2$.

The similar analysis of the curves in Figure 3.18 is in favor $\Gamma = 1$ to be accepted as the best value for the "Cameraman" test-image. The performance of the algorithm with $\Gamma = 1$ and $\Gamma = 1.5$ for "Cameraman" are shown in Figures 3.19-3.22. The adaptive window sizes are extremely noisy for $\Gamma = 1$ in Figure 3.20 as compared even with ones shown in Figure 3.22 for $\Gamma = 1.5$. Comparison of the final estimates of the "Cameraman" image in Figures 3.19 and 3.21 visually is in favor of Figure 3.21 for $\Gamma = 1.5$, while the criteria values are a bit better for $\Gamma = 1$. The multiple sports in Figure 3.19 are quite unpleasant. It provides an example of a situation, where the visual perception contradicts to the considered quantitative performance criteria.

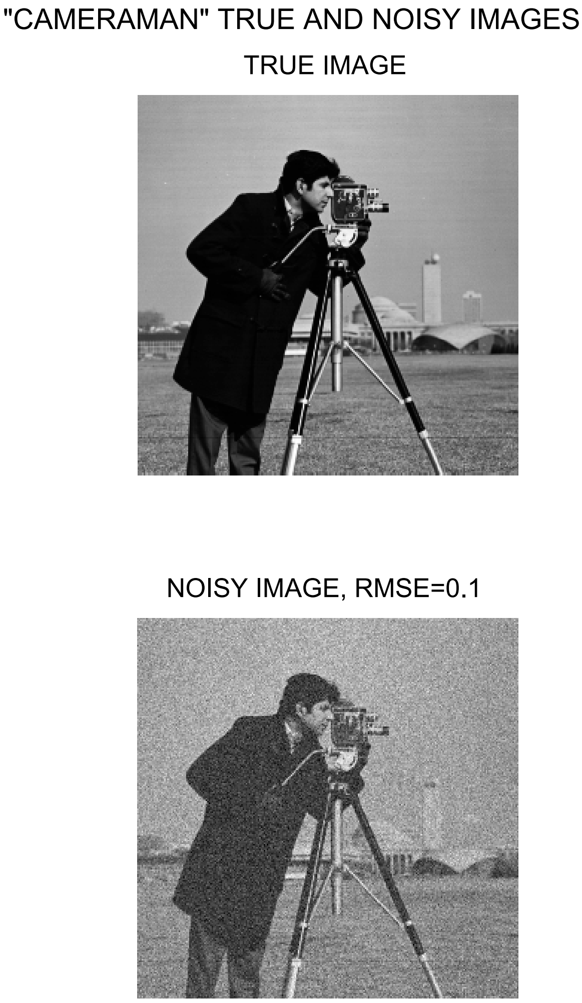


Figure 3.10: True and noisy image ”Cameraman”

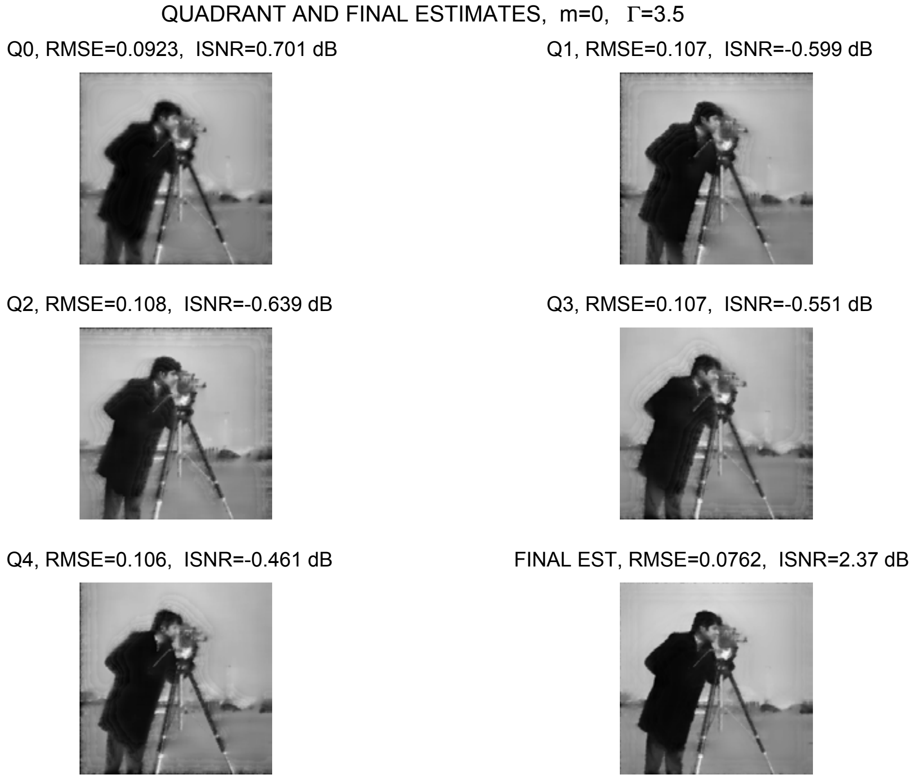

QUADRANT AND FINAL ESTIMATES, m=0, Γ=3.5
Q0, RMSE=0.0923, ISNR=0.701 dB
Q1, RMSE=0.107, ISNR=-0.599 dB
Q2, RMSE=0.108, ISNR=-0.639 dB
Q3, RMSE=0.107, ISNR=-0.551 dB
Q4, RMSE=0.106, ISNR=-0.461 dB
FINAL EST, RMSE=0.0762, ISNR=2.37 dB

Figure 3.11: "Cameraman" images show image estimates. The estimates are given for symmetric window estimate (Q_0), nonsymmetric quadrant window estimates (Q_1,Q_2,Q_3,Q_4), and for the final estimate formed from the quadrant and symmetric window estimates. $RMSE$ and $ISNR$ specify the accuracy of the corresponding estimates.

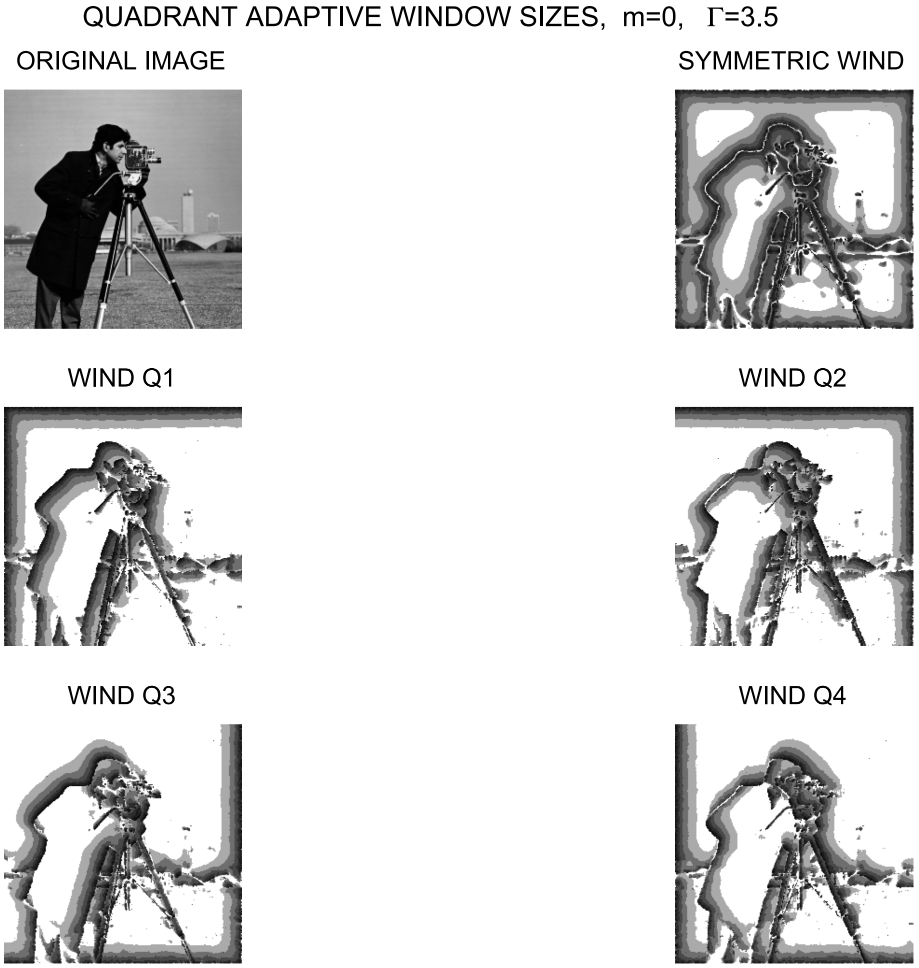
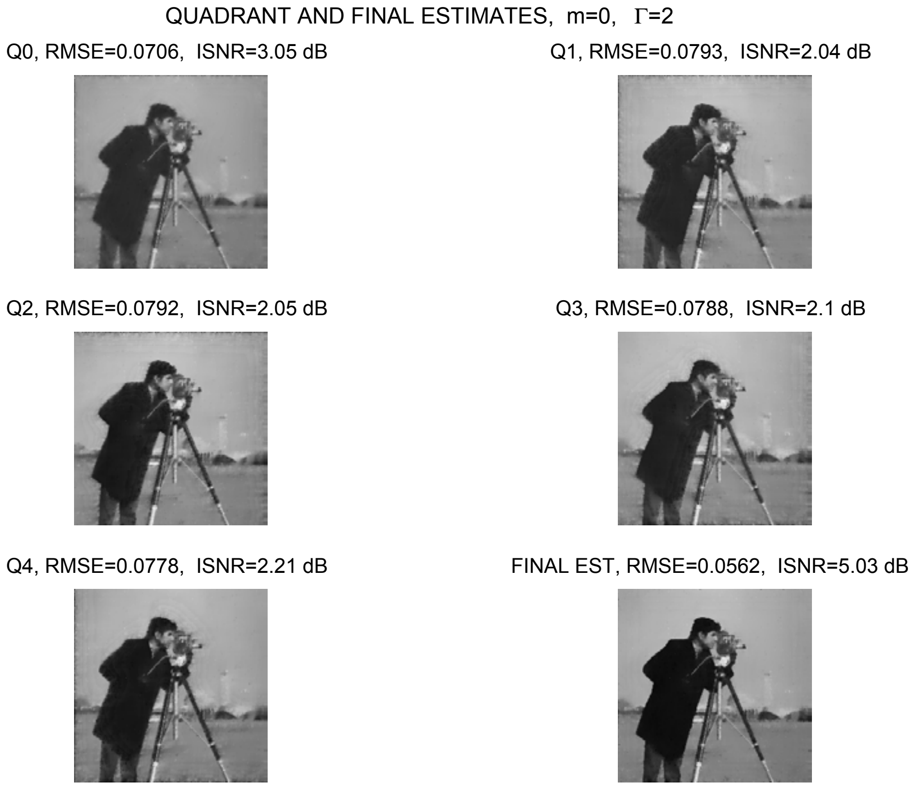


Figure 3.12: Cameraman images show the true image and adaptive varying window sizes obtained by the *ICI* rule for the symmetric window and quadrant image estimates.



QUADRANT AND FINAL ESTIMATES, m=0, Γ=2

Q0, RMSE=0.0706, ISNR=3.05 dB

Q1, RMSE=0.0793, ISNR=2.04 dB

Q2, RMSE=0.0792, ISNR=2.05 dB

Q3, RMSE=0.0788, ISNR=2.1 dB

Q4, RMSE=0.0778, ISNR=2.21 dB

FINAL EST, RMSE=0.0562, ISNR=5.03 dB

Figure 3.13: "Cameraman" image estimates. The estimates are given for the symmetric window estimate (Q_0), the nonsymmetric quadrant estimates (Q_1,Q_2,Q_3,Q_4), and for the final estimate formed from the quadrant and symmetric window estimates. *RMSE* and *ISNR* specify the accuracy of the corresponding estimates.

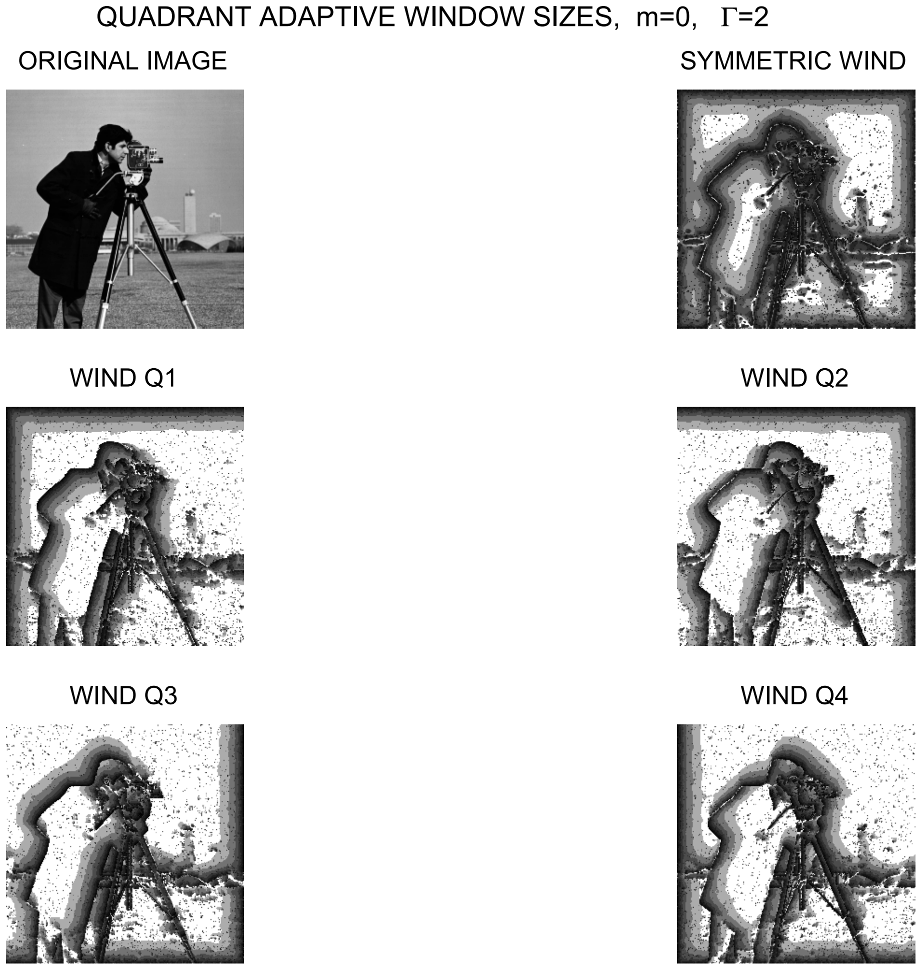


Figure 3.14: "Cameraman" true image and adaptive varying window sizes obtained by the *ICI* rule for the symmetric window and the quadrant estimates. Black spots are spikes, i.e. erroneous isolated small window size values.

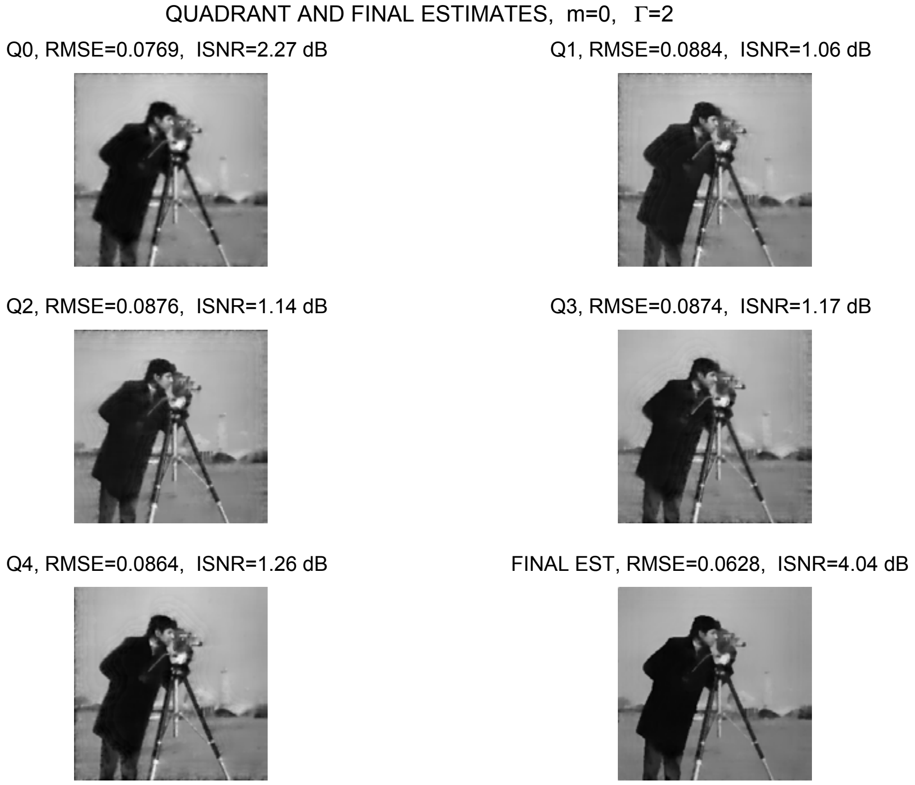

QUADRANT AND FINAL ESTIMATES, m=0, Γ=2
Q0, RMSE=0.0769, ISNR=2.27 dB
Q1, RMSE=0.0884, ISNR=1.06 dB
Q2, RMSE=0.0876, ISNR=1.14 dB
Q3, RMSE=0.0874, ISNR=1.17 dB
Q4, RMSE=0.0864, ISNR=1.26 dB
FINAL EST, RMSE=0.0628, ISNR=4.04 dB

Figure 3.15: "Cameraman" image. Prefiltering of the adaptive window sizes obtained by the ICI rule. The estimates are given for the symmetric window estimate (Q_0), the nonsymmetric quadrant estimates (Q_1,Q_2,Q_3,Q_4), and for the final estimate formed from the quadrant and symmetric window estimates obtained after prefiltering the ICI adaptive window sizes. $2D$ median filter 3×3 is used for the prefiltering. $RMSE$ and $ISNR$ specify the accuracy of the corresponding estimates.

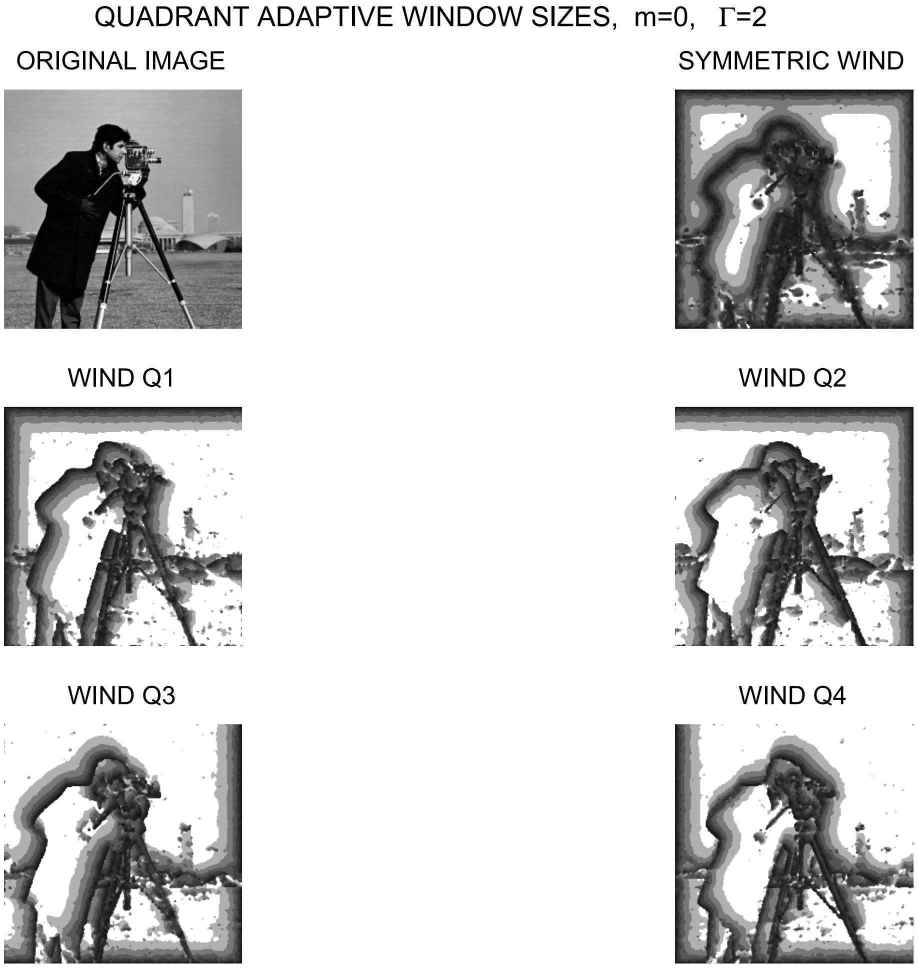


Figure 3.16: Filtered adaptive window sizes obtained by the ICI rule. The varying windows are given for the symmetric window estimate (Q_0), the nonsymmetric quadrant window estimates (Q_1,Q_2,Q_3,Q_4). $2D$ median filter 3×3 is used for this filtering. $RMSE$ and $ISNR$ specify the accuracy of the corresponding estimates.

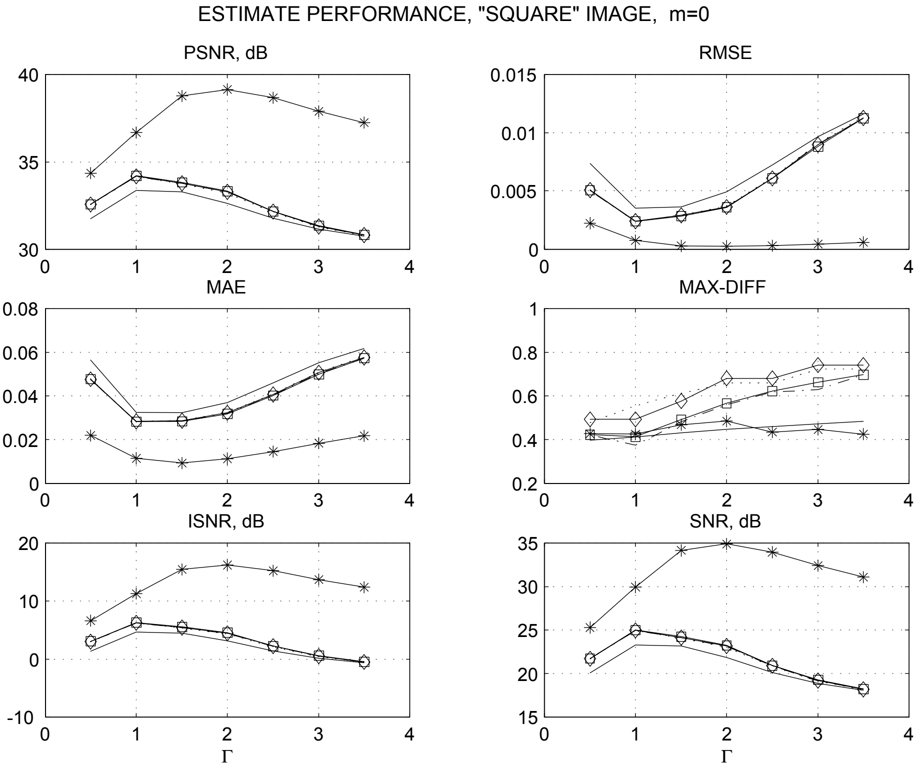


Figure 3.17: Performance criteria for "Square" image as functions of the threshold Γ. Criteria are given for the symmetric, quadrant and final estimates. Notation used: solid line for the symmetric window estimate, stars * for the final estimate, dotted lines, dash-dot lines, lines with squares and diamonds for the Q_1, Q_2, Q_3, and Q_4 quadrant estimates respectively.

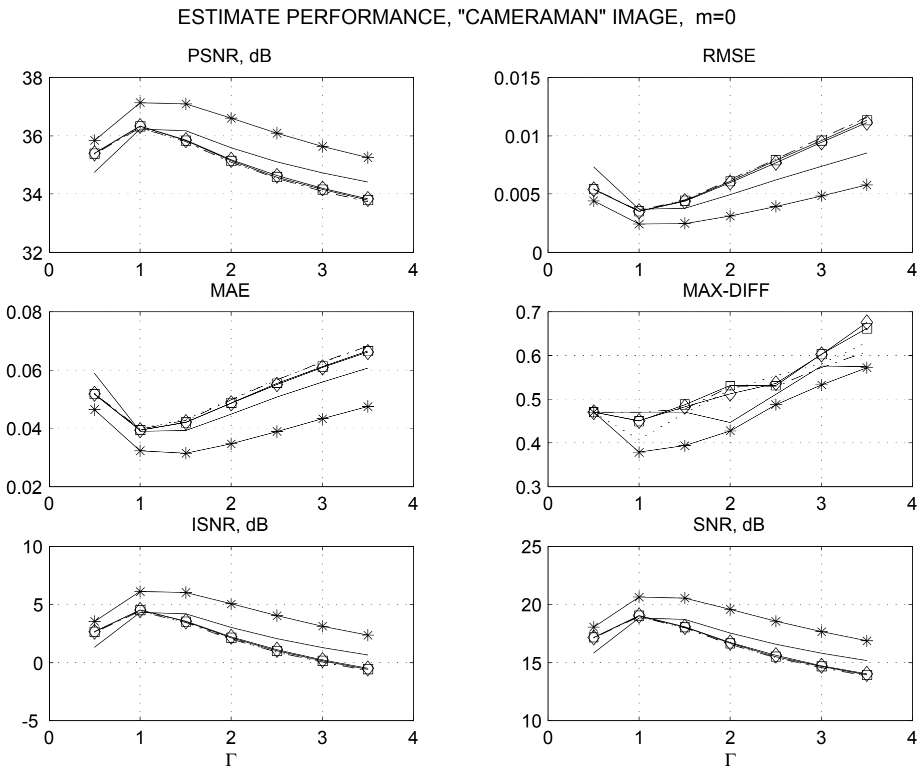


Figure 3.18: Performance criteria for "Cameraman" image as functions of the threshold Γ. Criteria are given for the symmetric, quadrant and final estimates. Notation used: solid line for the symmetric window estimate, stars * for the final estimate, dotted lines, dash-dot lines, lines with squares and diamonds for the Q_1, Q_2, Q_3, and Q_4 quadrant estimates respectively.

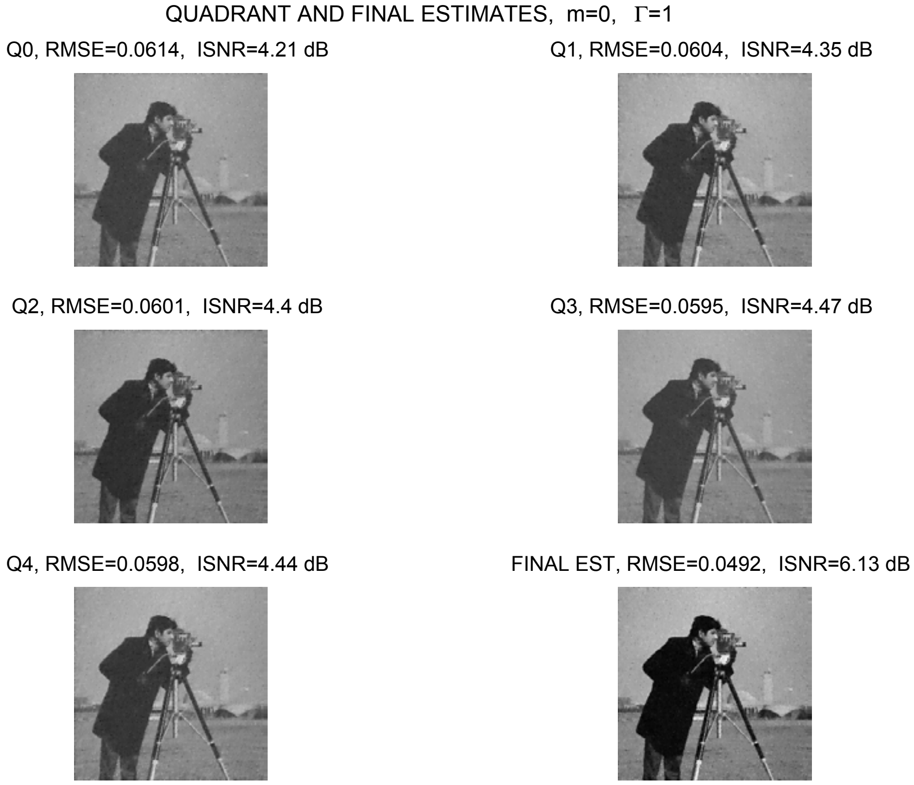

QUADRANT AND FINAL ESTIMATES, m=0, Γ=1
Q0, RMSE=0.0614, ISNR=4.21 dB
Q1, RMSE=0.0604, ISNR=4.35 dB
Q2, RMSE=0.0601, ISNR=4.4 dB
Q3, RMSE=0.0595, ISNR=4.47 dB
Q4, RMSE=0.0598, ISNR=4.44 dB
FINAL EST, RMSE=0.0492, ISNR=6.13 dB

Figure 3.19: "Cameraman" image estimates, $m = 0$ and $\Gamma = 1$. The estimates are given for symmetric window estimate (Q_0), nonsymmetric quadrant window estimates (Q_1,Q_2,Q_3,Q_4), and for the final estimate formed from the quadrant and symmetric window estimates. $RMSE$ and $ISNR$ specify the accuracy of the corresponding estimates.

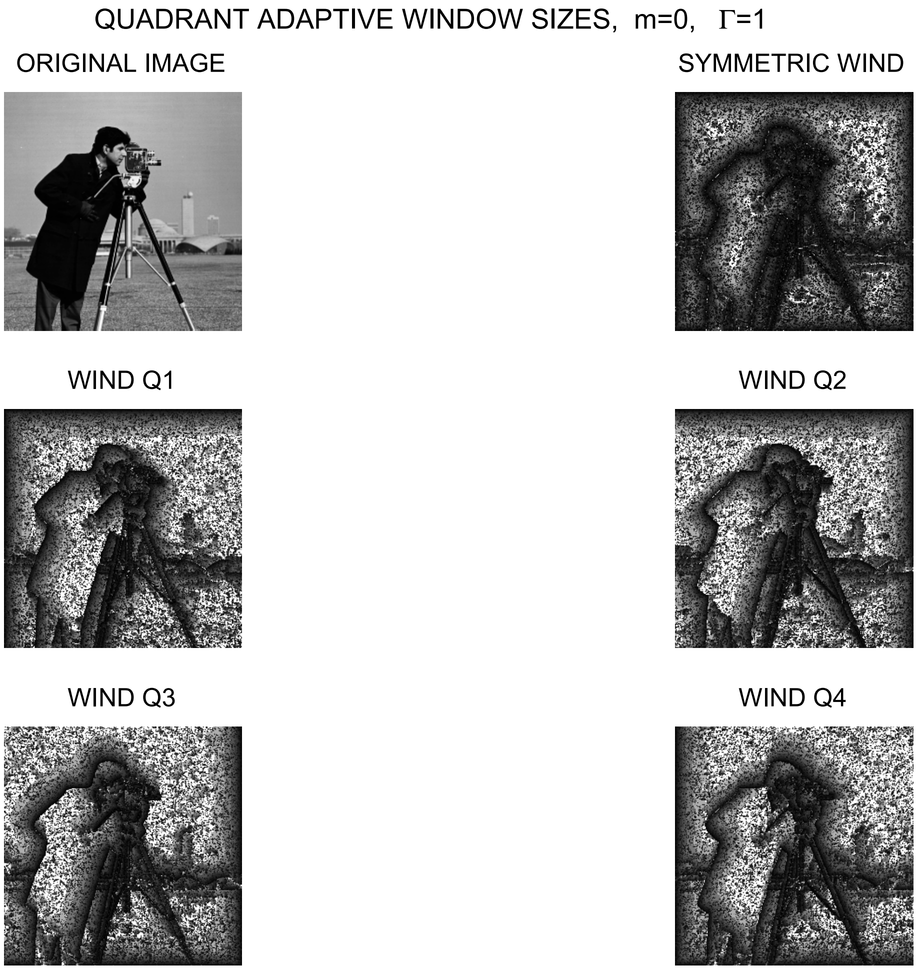


Figure 3.20: The ICI adaptive window sizes for "Cameraman" image estimation with the optimal $\Gamma = 1$. The adaptive window sizes are extremely noisy.

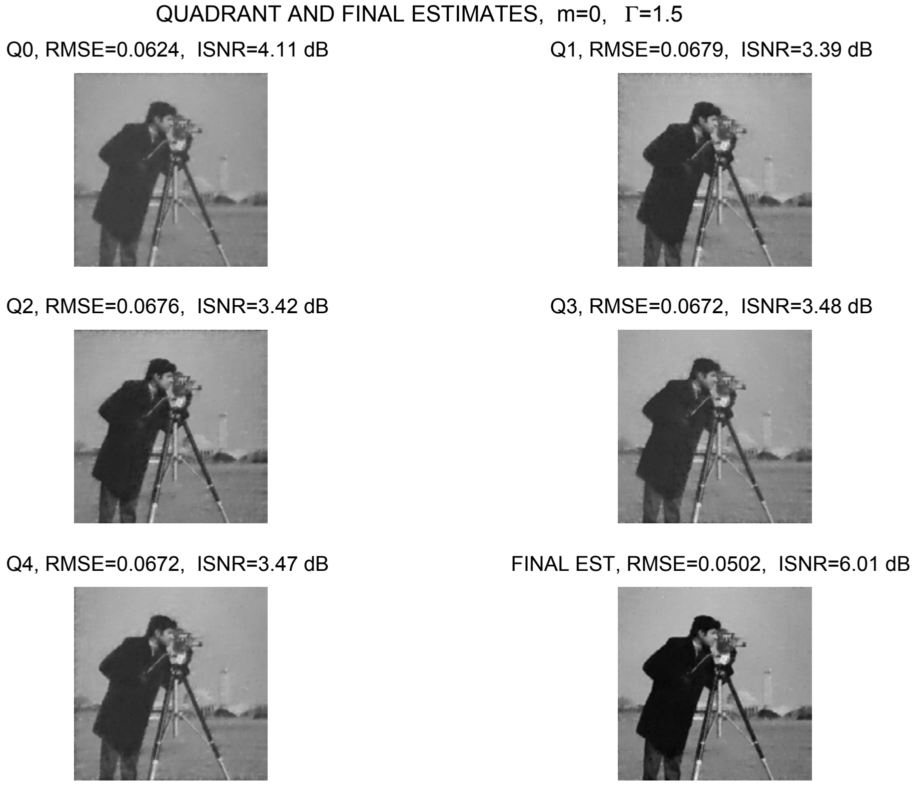
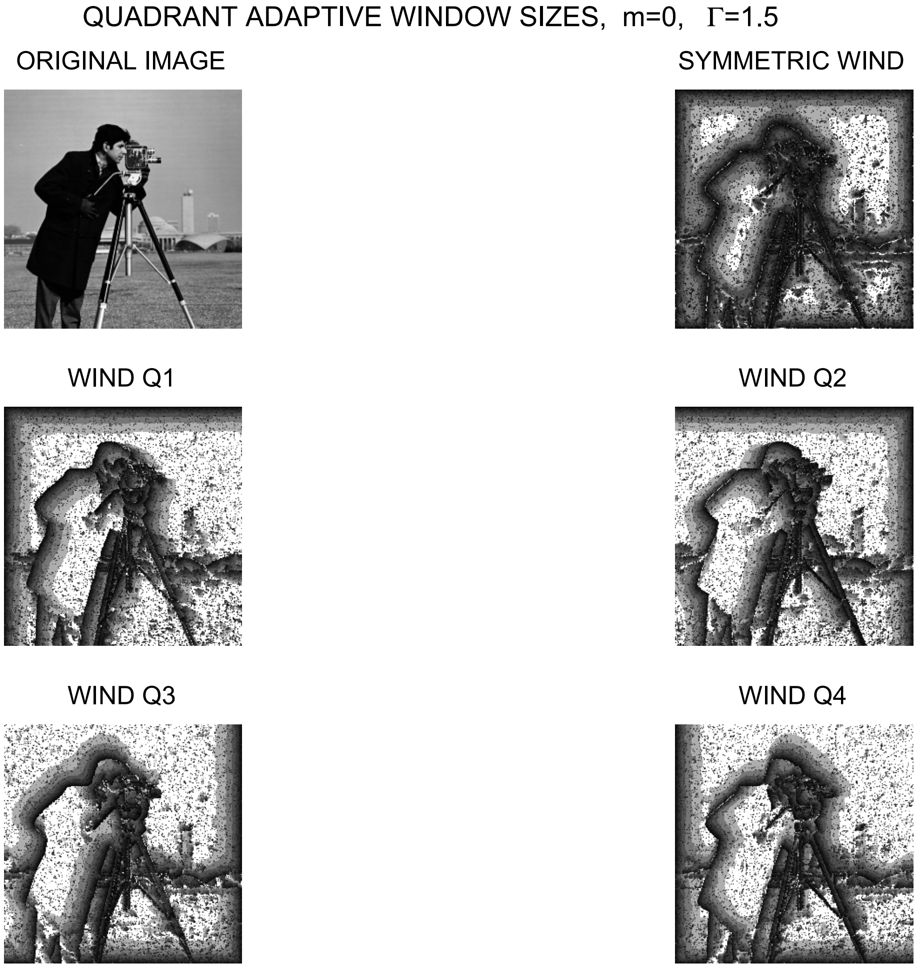

QUADRANT AND FINAL ESTIMATES, m=0, Γ=1.5
Q0, RMSE=0.0624, ISNR=4.11 dB
Q1, RMSE=0.0679, ISNR=3.39 dB
Q2, RMSE=0.0676, ISNR=3.42 dB
Q3, RMSE=0.0672, ISNR=3.48 dB
Q4, RMSE=0.0672, ISNR=3.47 dB
FINAL EST, RMSE=0.0502, ISNR=6.01 dB

Figure 3.21: "Cameraman" image estimation for the nearly optimal $\Gamma = 1.5$. The value of Γ larger than the optimal $\Gamma = 1$ makes the estimate more acceptable from the point of view of the visual perception.



QUADRANT ADAPTIVE WINDOW SIZES, m=0, Γ=1.5

ORIGINAL IMAGE

SYMMETRIC WIND

WIND Q1

WIND Q2

WIND Q3

WIND Q4

Figure 3.22: These *ICI* adaptive window sizes are noisy. However, they are less noisy in comparison with similar imaging in Figure 3.20 given for the optimal threshold $\Gamma = 1$.

3.1.3 ICI-varying window sizes, $m = 1$ and $m = 2$

Let us use the higher power LPA estimates. It follows from the idea of the LPA and from the accuracy analysis that the higher power approximation is reasonable provided that the image intensity function has a corresponding smoothness, i.e. the corresponding higher order derivatives exist and small. The concept of smoothness can be easily formalized and tested for simple geometrical images, say such as "Square". However, for usual real images, for instance such as "Cameraman", there is no simple answers and tests on smoothness, as a situation is varying for different segments of the image.

In this section we use simulation in order to derive some general conclusions.

We start from the analysis the above six criteria (3.7)-(3.12) as functions of Γ and use these criteria for optimization of Γ.

Figure 3.23 and Figure 3.24 show these criteria for the test-images "Square" and for the LPA of the powers $m = 1$ and $m = 2$, respectively. It can be seen, that for $m = 1$ the optimal value $\Gamma = 3$. The criteria curves given in Figure 3.24 for $m = 2$ show a clear degradation of the filtering performance as compared with the case $m = 1$. It means, that the second power LPA definitely does not fit to approximation of the piece-wise "Square" image and larger values of m result in a significant increasing of the level of random components. It is an example, when using the higher power LPA worsens the estimates. However, comparison of Figure 3.23, $m = 1$, with the curves corresponding $m = 0$ in Figure 3.17 is definitely in favor of the first power LPA.

Figure 3.25 and Figure 3.26 show the criteria for the test-images "Cameraman" with the LPA of the powers $m = 1$ and $m = 2$, respectively. In the both cases the optimal value $\Gamma = 1.5$, while the criteria values are nearly identical for these powers. The estimates for $m = 1$ and $m = 2$ with the optimal $\Gamma = 1.5$ are demonstrated in Figures 3.29 and Figure 3.30. Comparison of these results with ones achieved for $m = 0$ and shown in Figure 3.18 are clearly in favor of the higher power estimates. The values of the criteria $RMSE$ and $ISNR$ are slightly in favor estimation $m = 1$, however, it looks like the visual perception is in favor of the higher power LPA with $m = 2$.

For comparison, we show also the estimates for "Cameraman" obtained for $m = 1$ and $m = 2$ for $\Gamma = 2$, Figures 3.31 and 3.32 This Γ is not optimal, and we can observe a difference between the values of $RMSE$ and $ISNR$. However, the visual effects are in favor of the non-optimal larger $\Gamma = 2$.

Let us make some general conclusions from the discussed simulations:

(1) The six performance criteria considered for Γ optimization achieve extremum values at the same values of Γ. It allows to use only some of these criteria. Further we will consider only criteria $RMSE$ and $ISNR$ as the most simple and appealing;

(2) The reasonable values of Γ belong to the interval $[1, 3]$.

(3) The higher power LPA, $m > 0$, is able to improve the performance in quite a significant way. In particular, for the binary "Square" image the best results are obtained with $m = 1$. For the gray-scale image "Cameraman" criteria show the best values for $m = 1$, while the visual perception is in favor of $m = 2$.

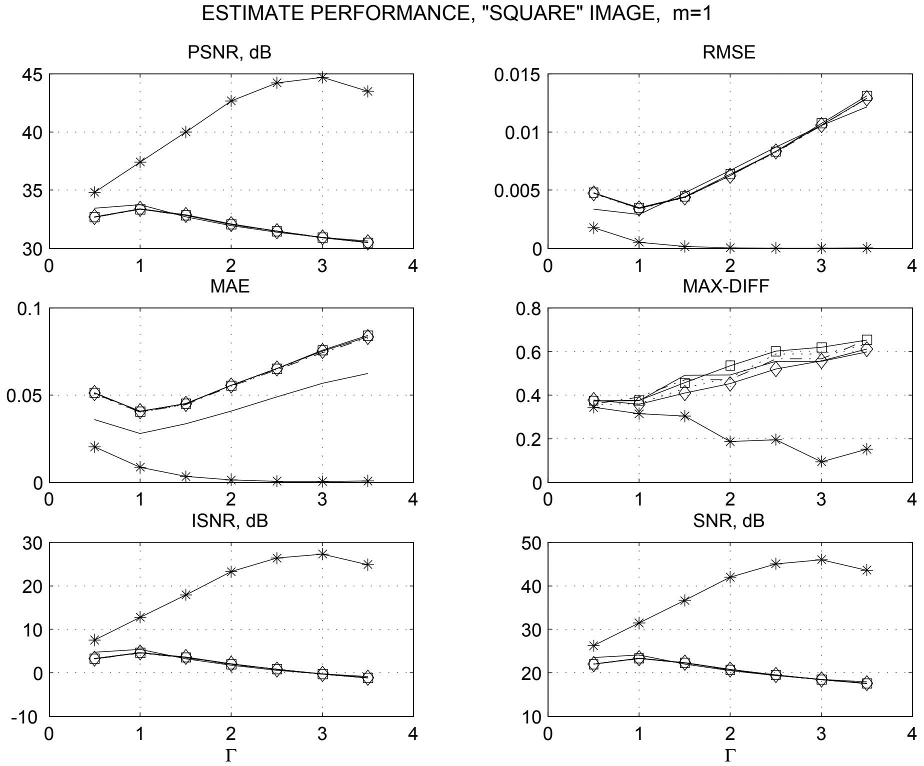


Figure 3.23: Performance criteria for "Square" image as functions of the threshold Γ, $m = 1$. Criteria are given for the symmetric, quadrant and final estimates. Notation used: solid line for the symmetric window estimate, stars * for the final estimate, dotted lines, dash-dot lines, lines with squares and diamonds for the Q_1, Q_2, Q_3, and Q_4 quadrant estimates respectively.

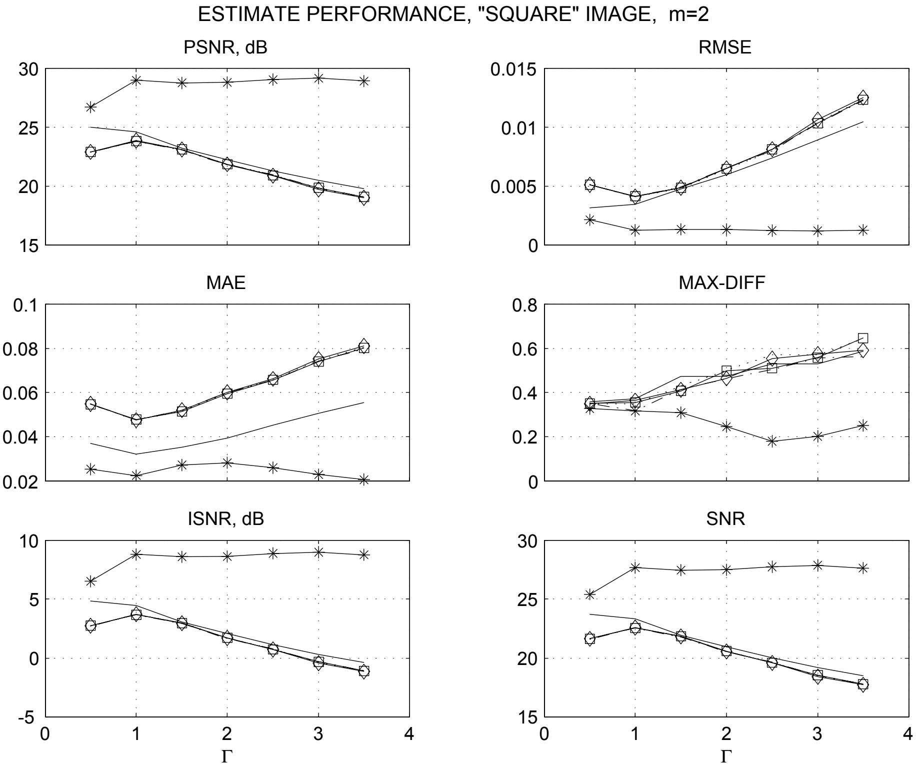


Figure 3.24: Performance criteria for "Square" image as functions of the threshold Γ, $m = 2$. Criteria are given for the symmetric, quadrant and final estimates. Notation used: solid line for the symmetric window estimate, stars * for the final estimate, dotted lines, dash-dot lines, lines with squares and diamonds for the Q_1, Q_2, Q_3, and Q_4 quadrant estimates, respectively.

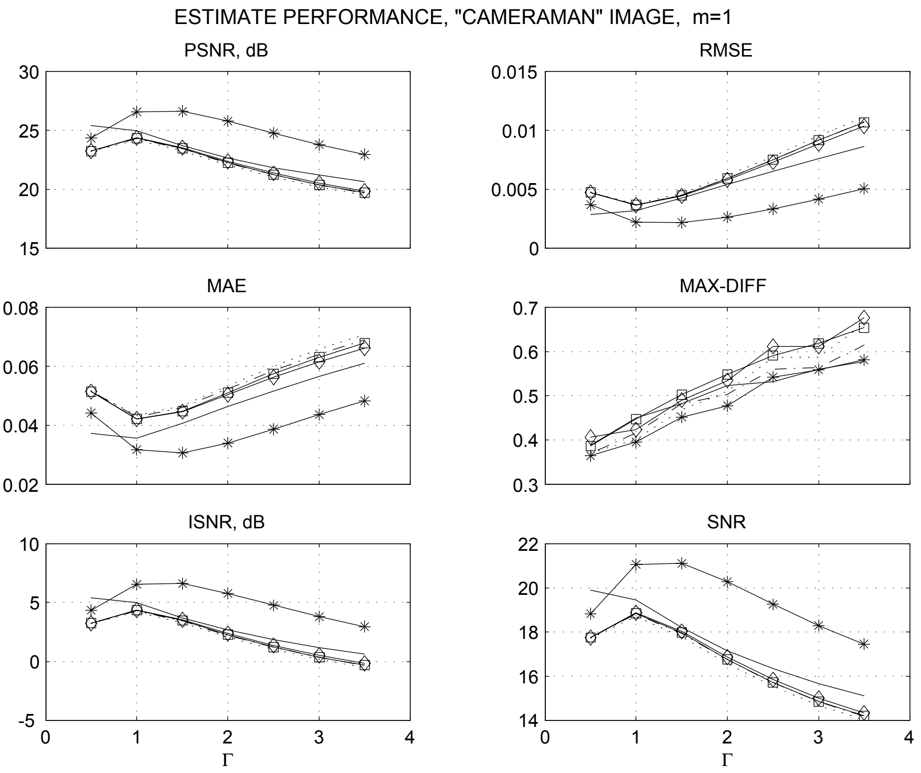


Figure 3.25: Performance criteria for "Cameraman" image as functions of the threshold Γ, $m = 1$. Criteria are given for the symmetric, quadrant and final estimates. Notation used: solid line for the symmetric window estimate, stars * for the final estimate, dotted lines, dash-dot lines, lines with squares and diamonds for the Q_1, Q_2, Q_3, and Q_4 quadrant estimates, respectively.

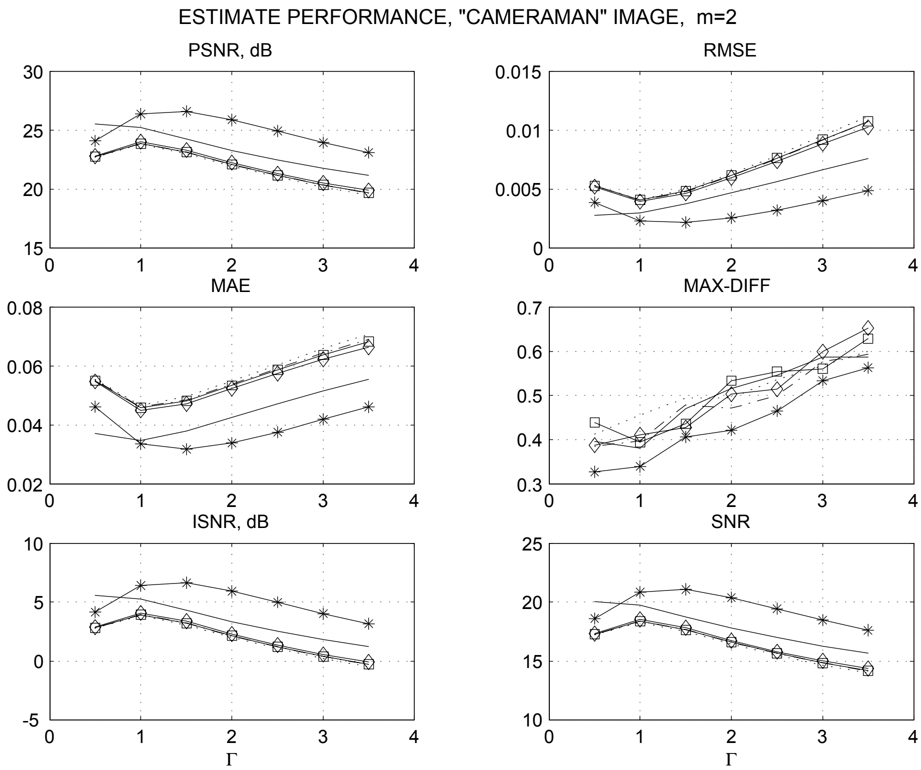


Figure 3.26: Performance criteria for "Cameraman" image as functions of the threshold Γ, $m = 2$. Criteria are given for the symmetric, quadrant and final estimates. Notation used: solid line for the symmetric window estimate, stars * for the final estimate, dotted lines, dash-dot lines, lines with squares and diamonds for the Q_1, Q_2, Q_3, and Q_4 quadrant estimates, respectively.

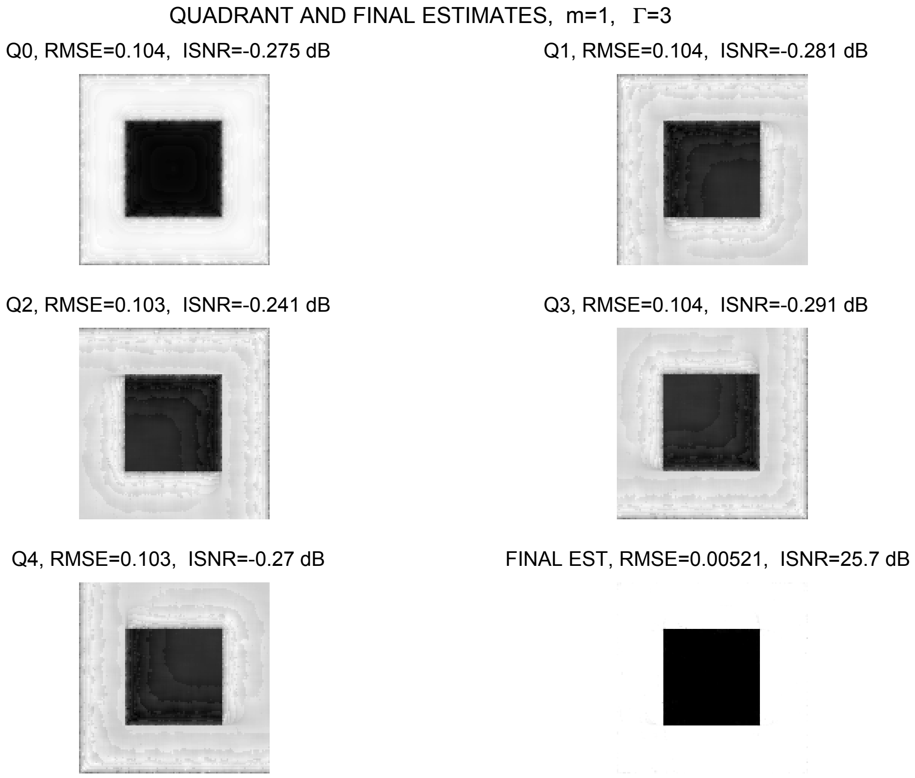

QUADRANT AND FINAL ESTIMATES, m=1, Γ=3
Q0, RMSE=0.104, ISNR=-0.275 dB
Q1, RMSE=0.104, ISNR=-0.281 dB
Q2, RMSE=0.103, ISNR=-0.241 dB
Q3, RMSE=0.104, ISNR=-0.291 dB
Q4, RMSE=0.103, ISNR=-0.27 dB
FINAL EST, RMSE=0.00521, ISNR=25.7 dB

Figure 3.27: "Square" image estimates, $m = 1$. The estimates are given for the symmetric window estimate (Q_0), the nonsymmetric quadrant window estimates (Q_1,Q_2,Q_3,Q_4), and for the final estimates formed from the quadrant and symmetric window estimates. $RMSE$ and $ISNR$ specify the accuracy of the corresponding estimates.

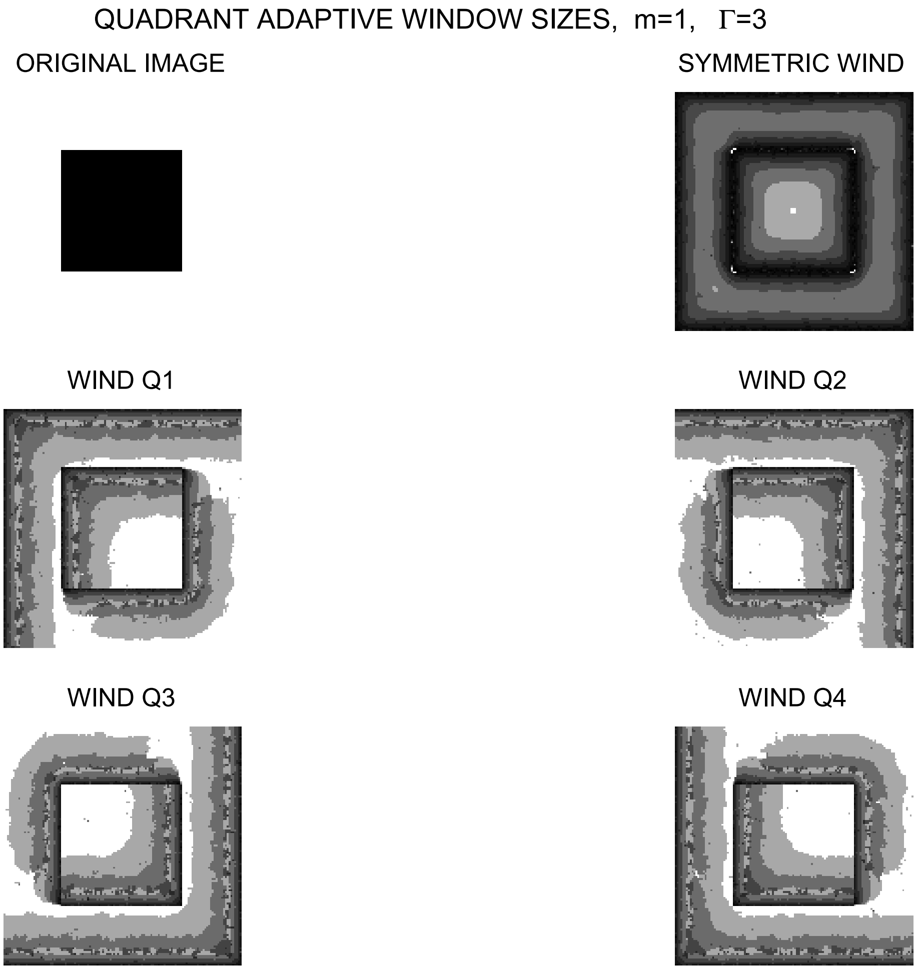
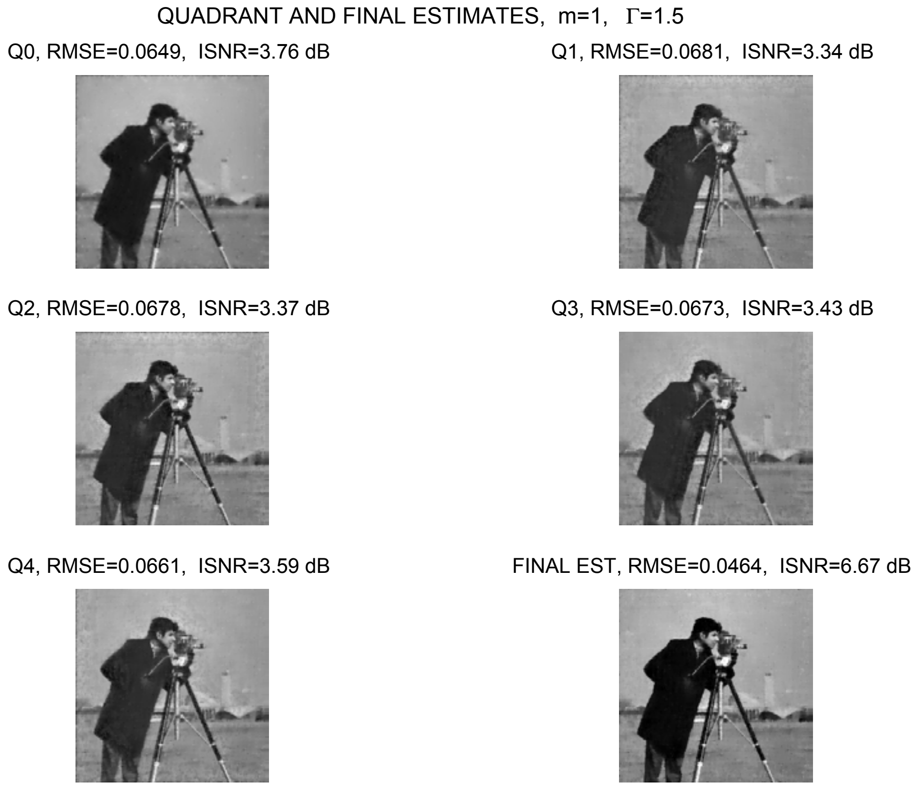


Figure 3.28: "Square" true image and adaptive varying window sizes obtained by the ICI, $m = 1$ and $\Gamma = 3$ for the symmetric window and quadrant image estimates.



QUADRANT AND FINAL ESTIMATES, m=1, Γ=1.5

Q0, RMSE=0.0649, ISNR=3.76 dB

Q1, RMSE=0.0681, ISNR=3.34 dB

Q2, RMSE=0.0678, ISNR=3.37 dB

Q3, RMSE=0.0673, ISNR=3.43 dB

Q4, RMSE=0.0661, ISNR=3.59 dB

FINAL EST, RMSE=0.0464, ISNR=6.67 dB

Figure 3.29: "Cameraman" image estimates for $m = 1$ and the optimal $\Gamma = 1.5$.

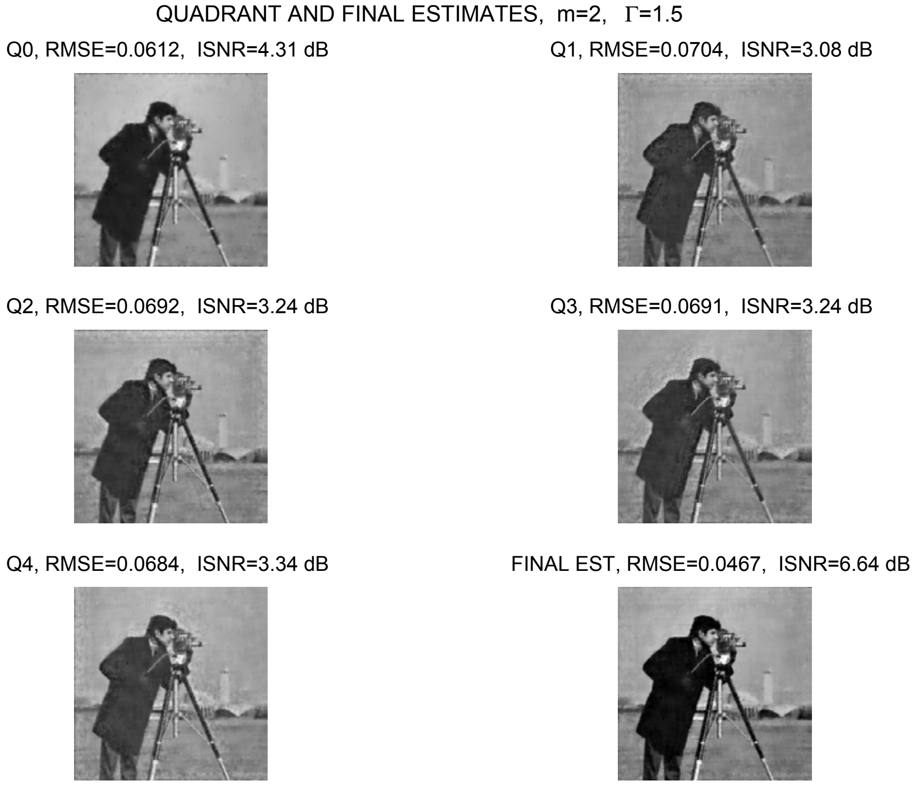

QUADRANT AND FINAL ESTIMATES, m=2, Γ=1.5
Q0, RMSE=0.0612, ISNR=4.31 dB
Q1, RMSE=0.0704, ISNR=3.08 dB
Q2, RMSE=0.0692, ISNR=3.24 dB
Q3, RMSE=0.0691, ISNR=3.24 dB
Q4, RMSE=0.0684, ISNR=3.34 dB
FINAL EST, RMSE=0.0467, ISNR=6.64 dB

Figure 3.30: "Cameraman" image estimation for $m = 2$ and the optimal $\Gamma = 1.5$.

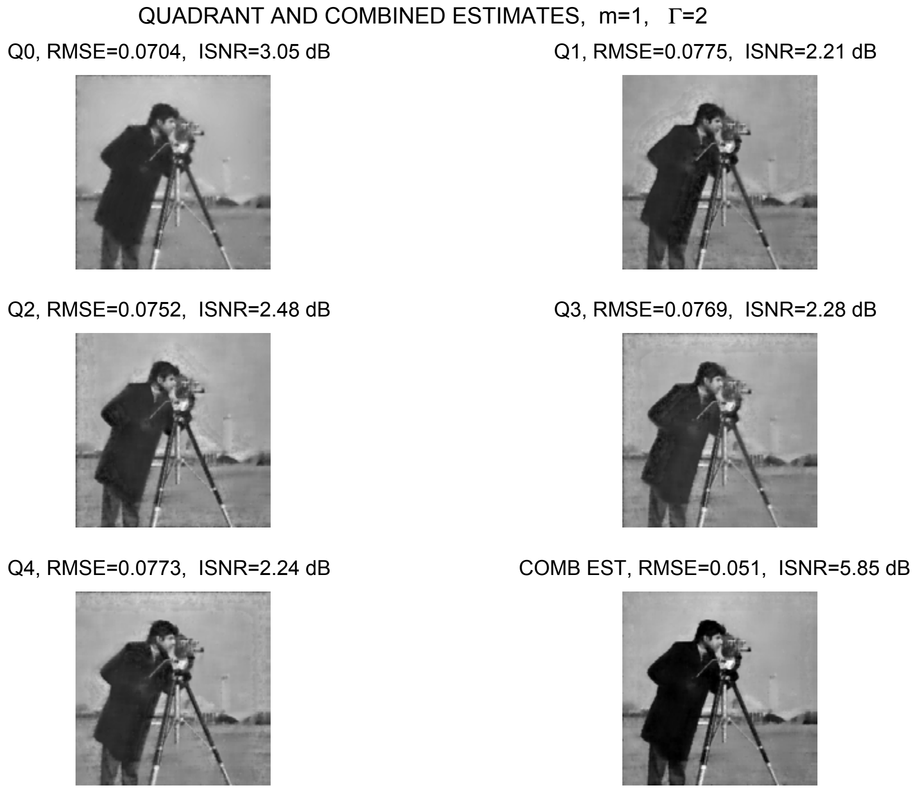

QUADRANT AND COMBINED ESTIMATES, m=1, Γ=2
Q0, RMSE=0.0704, ISNR=3.05 dB
Q1, RMSE=0.0775, ISNR=2.21 dB
Q2, RMSE=0.0752, ISNR=2.48 dB
Q3, RMSE=0.0769, ISNR=2.28 dB
Q4, RMSE=0.0773, ISNR=2.24 dB
COMB EST, RMSE=0.051, ISNR=5.85 dB

Figure 3.31: Cameraman images show image estimates. The estimates are given for the symmetric window estimate (Q_0), the nonsymmetric quadrant window estimates (Q_1,Q_2,Q_3,Q_4), and for the final combined estimates formed from the quadrant and symmetric window estimates. $RMSE$ and $ISNR$ specify the accuracy of the corresponding estimates.

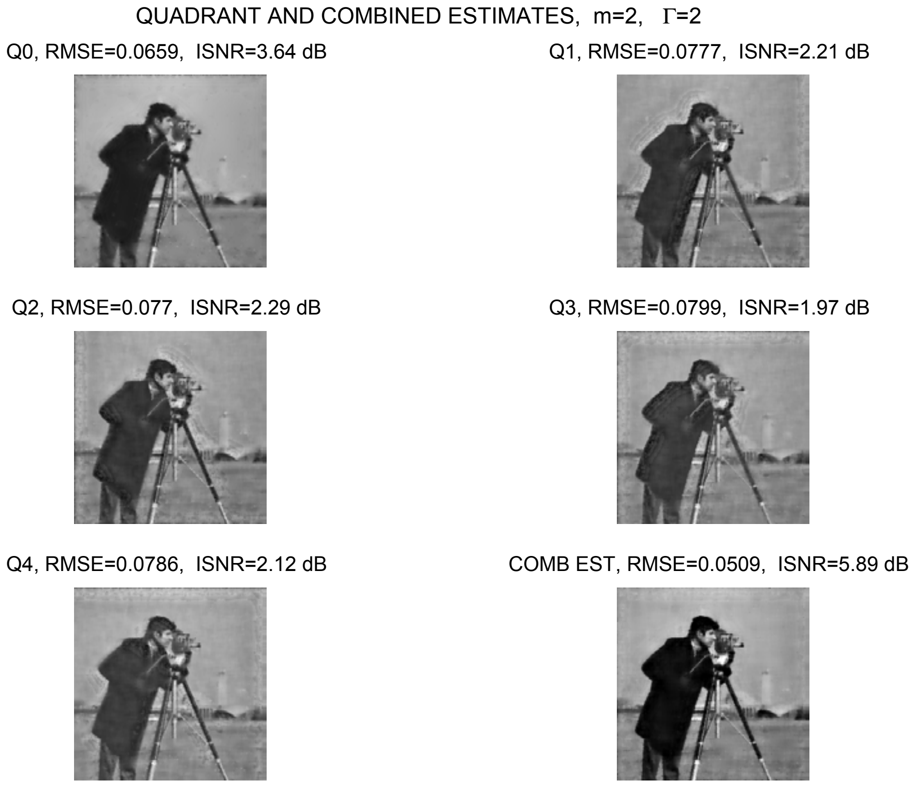

QUADRANT AND COMBINED ESTIMATES, m=2, Γ=2
Q0, RMSE=0.0659, ISNR=3.64 dB
Q1, RMSE=0.0777, ISNR=2.21 dB
Q2, RMSE=0.077, ISNR=2.29 dB
Q3, RMSE=0.0799, ISNR=1.97 dB
Q4, RMSE=0.0786, ISNR=2.12 dB
COMB EST, RMSE=0.0509, ISNR=5.89 dB

Figure 3.32: Cameraman images show image estimates. The estimates are given for the symmetric window estimate (Q_0), the nonsymmetric quadrant window estimates (Q_1,Q_2,Q_3,Q_4), and for the final combined estimates formed from the quadrant and symmetric window estimates. $RMSE$ and $ISNR$ specify the accuracy of the corresponding estimates. Using the LPA of the power $m = 2$ gives the quality of filtering close to achieved for $m = 1$.

3.2 Differentiation

When the power m of the LPA is equal or higher than 1 then the LPA estimates of the derivatives can be calculated. All these estimates can be used also with varying adaptive window sizes and this topic is of the main concern in this section.

We differentiate two different ideas exploited for adaptive varying scale derivative estimation:

(1) Function-ICI estimates;

(2) Derivative-ICI estimates.

The former assumes that the adaptive window sizes are obtained for the function (intensity) estimation and used for the derivative estimation. In this case we apply the same window sizes for simultaneous function and derivative estimation.

The second idea assumes that the ICI rule is applied to the derivative estimates and these derivative estimates use the window sizes different from those for the function estimation.

Actually, it makes a big deal of difference with the function-ICI adaptive estimates. In particular, it means that different window sizes appear in estimation of different derivatives. The algorithms become more bulky, while they are able to improve a performance of the derivative estimators.

Here we give some illustration of the adaptive scale differentiation.

3.2.1 Function-ICI derivative estimation

The function-ICI derivative estimates for the test-images "Square" and " Cameraman" with $m = 1$ are shown in Figures 3.33, 3.34, 3.35, 3.36. These estimates are obtained for the symmetric and quadrant windows w_h. The corresponding quadrant estimates as well as the optimal window sizes of "Square" image can be seen in Figure 3.27 and Figure 3.28. They are produced for the optimal $\Gamma = 3$. The quadrant estimates of the "Cameraman" image corresponding to the results shown in Figures 3.35 and 3.36 can be seen Figure in 3.29. These estimates are given for the optimal $\Gamma = 1.5$. Note that the quadrant nonsymmetric derivative estimates are actually directional with the directions defined by the used quadrant. Visually, these derivative estimates are sin a good agreement with what it could be expected.

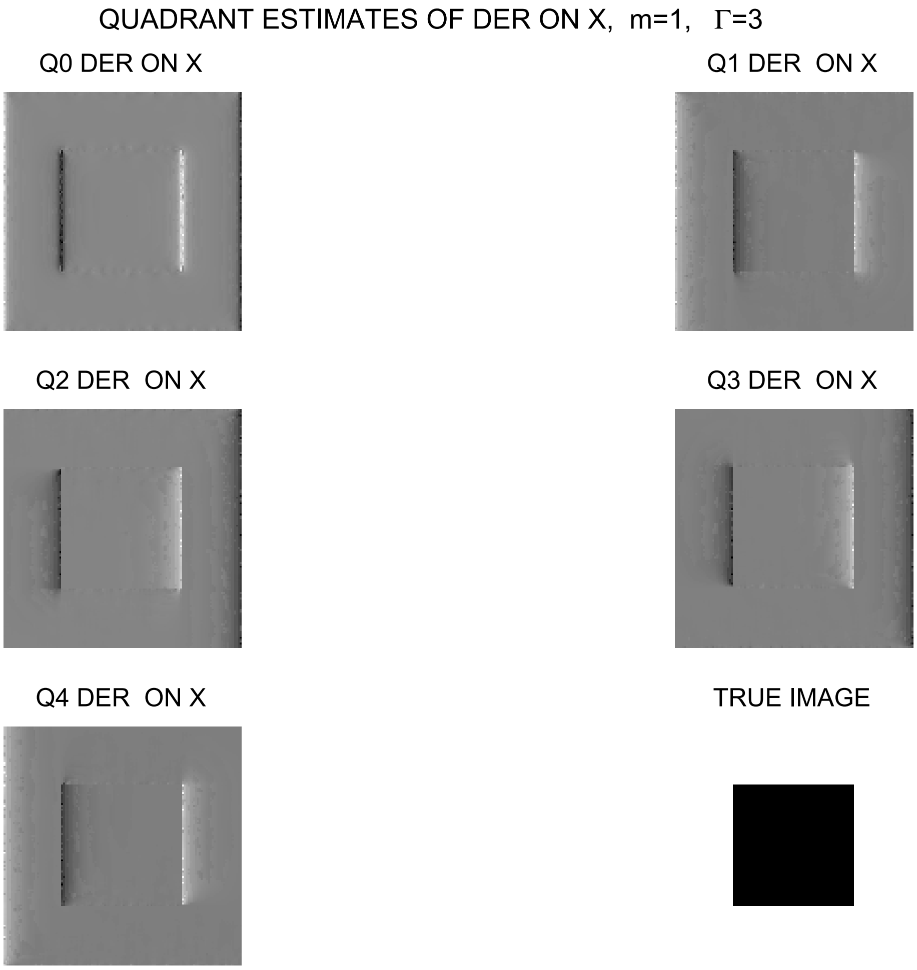


Figure 3.33: Quadrant estimates of the derivative on X, "Square" image. The adaptive window sizes are found for the intensity estimation, i.e. the function-ICI adaptive algorithm is used.

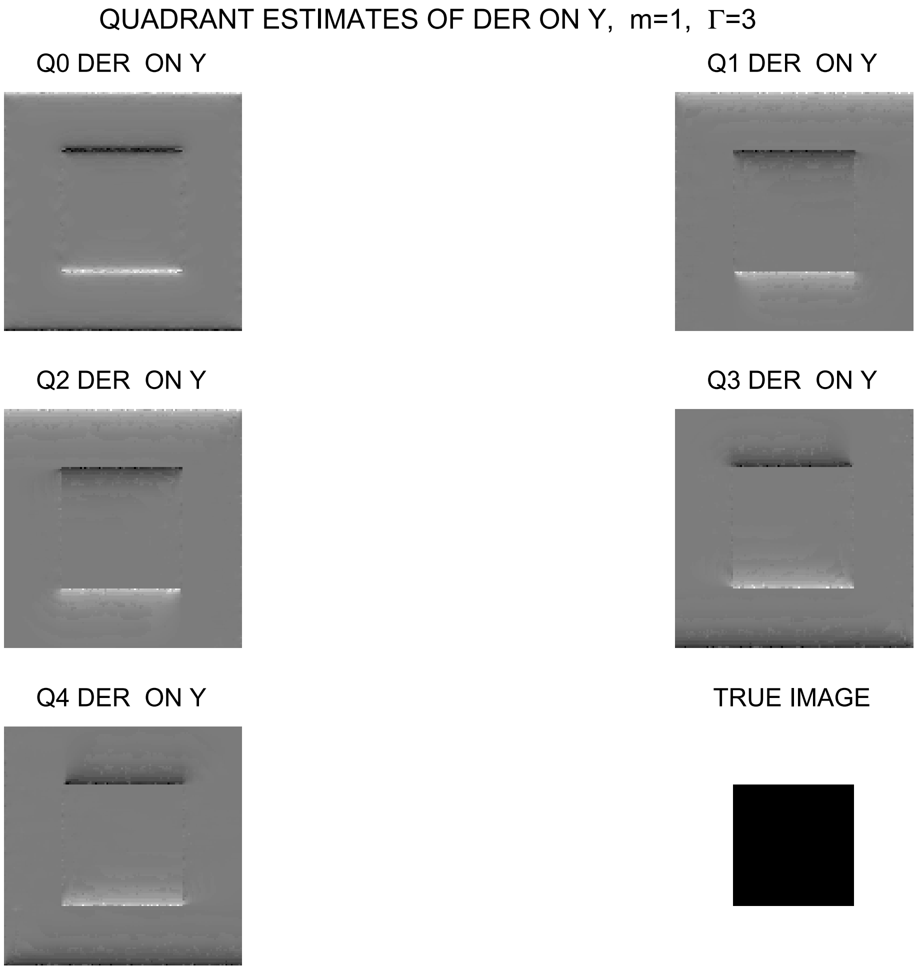


Figure 3.34: Quadrant estimates of the derivative on Y, "Square" image. The adaptive window sizes are found for the intensity estimation, i.e. the function-ICI adaptive algorithm is used.

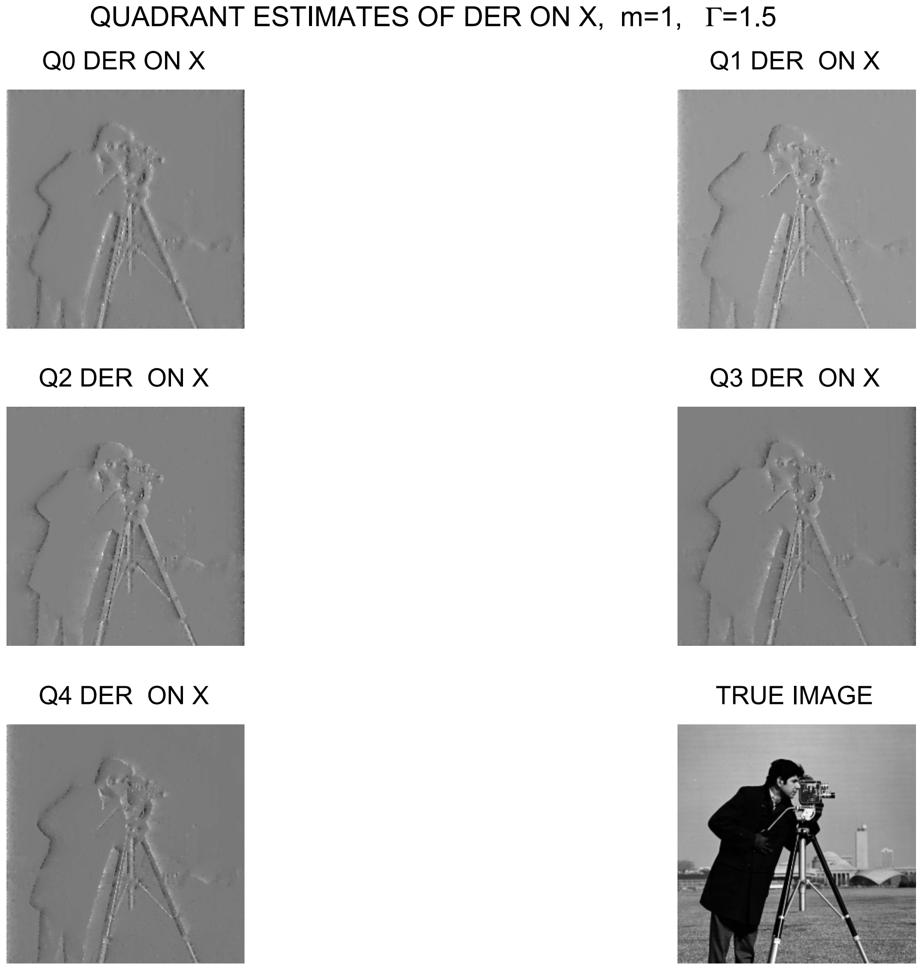


Figure 3.35: Quadrant estimates the derivative on X, "Cameraman" image. The adaptive window sizes are found for the intensity estimation, i.e. the function-ICI adaptive algorithm is used.

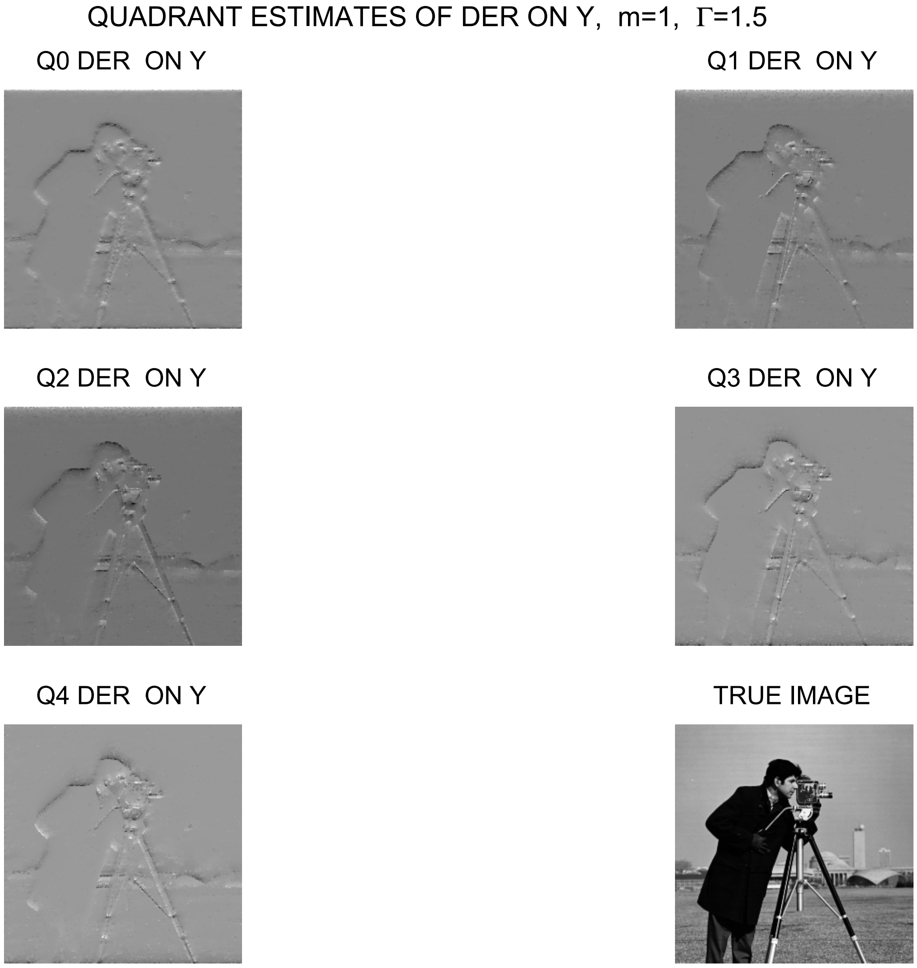


Figure 3.36: Quadrant estimates of the derivatives on Y , "Cameraman" image. The adaptive window sizes are found for the intensity estimation, i.e. the function-ICI adaptive algorithm is used.

3.2.2 Derivative-ICI derivative estimation

The derivative-ICI estimation assumes that the varying adaptive scale is obtained by applying the ICI rule to the estimates of the derivatives.

In this case, there are multiple possibilities as we are able to use the adaptive scale obtained, say for estimation of the derivatives on X, in order to estimate this derivative as well as all others, However, we are able to use the adaptive window sizes obtained for the derivative on Y in a similar way, in order to estimate this derivatives as well as all others.

The best performance naturally can be expected using the adaptive window sizes for estimation of the corresponding derivative only. We should also to keep in mind that there are also the symmetric and quadrants estimates, which are able to generate five types of the estimates for the every derivative. In this section we wish to give some illustrations of a variety of the possible estimates and adaptive scale selections available in terms of the developed approach in general.

In Figures 3.37 and 3.38 we may see the ICI adaptive window sizes for the estimates $\partial/\partial X$ and $\partial/\partial Y$ respectively. Comparison of the corresponding quadrant and symmetric window sizes show a difference of the windows obtained for estimation of the derivative in the horizontal and vertical directions. Thus, the adaptive windows are different for different derivative estimates.

Figures 3.39, 3.40, 3.41 and 3.42 show the adaptive estimates of the derivative $\partial/\partial X$, $\partial/\partial Y$ using the ICI adaptive window sizes obtained for the derivatives $\partial/\partial X$, $\partial/\partial Y$. We have here four combinations of the estimates and the possible ICI adaptive window sizes:

(1) The estimation of the derivative $\partial/\partial X$ using the adaptive windows obtained for this derivative $\partial/\partial X$ (Figure 3.39);

(2) The estimation of the derivative $\partial/\partial Y$ using the adaptive window obtained for another derivative $\partial/\partial X$ (Figure 3.40);

(3) The estimation of the derivative $\partial/\partial X$ using the adaptive window obtained for the derivative $\partial/\partial Y$ (Figure 3.41);

(4) The estimation of the derivative $\partial/\partial Y$ using the adaptive window obtained for this derivative $\partial/\partial Y$ (Figure 3.42).

Another aspect of the problem is illustrated in Figures 3.43 and 3.44, where we show the estimates of the image "Cameraman" using the ICI adaptive window sizes obtained for the derivatives. It can be conclude that these estimates are not good visually as well as according to the corresponding values of the criteria.

However, the window sizes obtained for the image intensity are quite acceptable for derivative estimates. One of the important advantage of this sort of adaptive derivative estimates is that the same adaptive window sizes are used for calculation the intensity as well as all its derivatives.

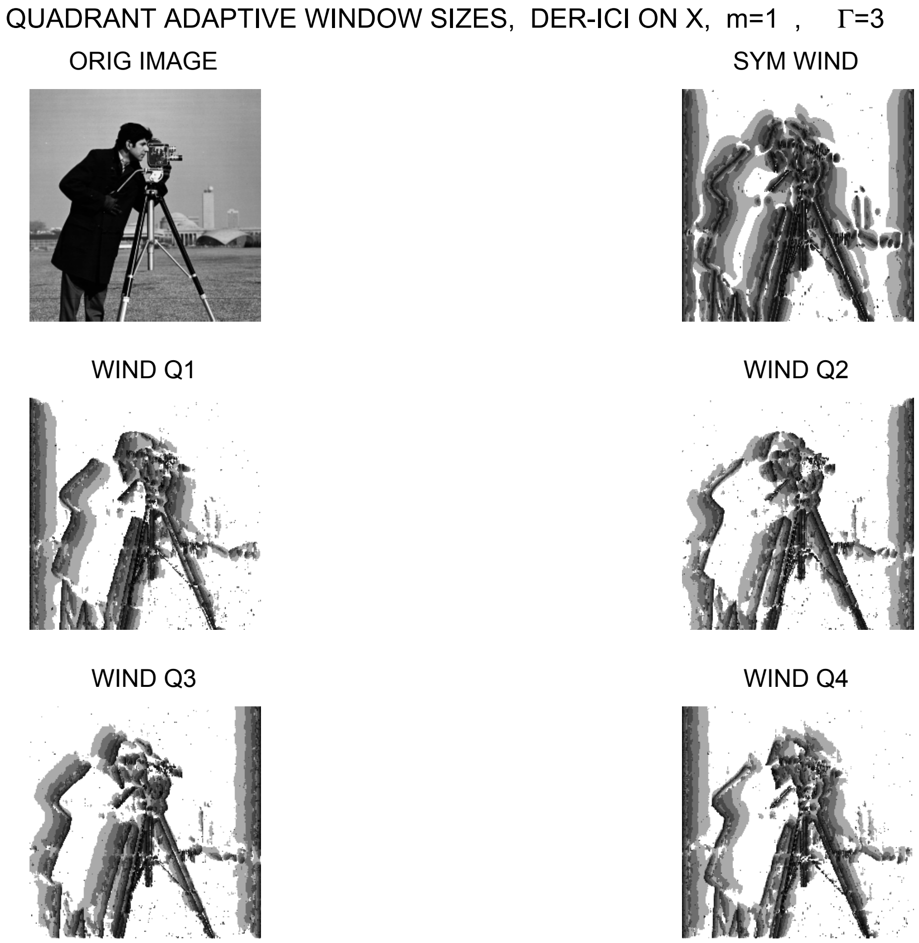


Figure 3.37: The *ICI* adaptive varying window sizes obtained for $\partial/\partial X$ derivative estimates. The window sizes are presented for the symmetric and quadrant windows.

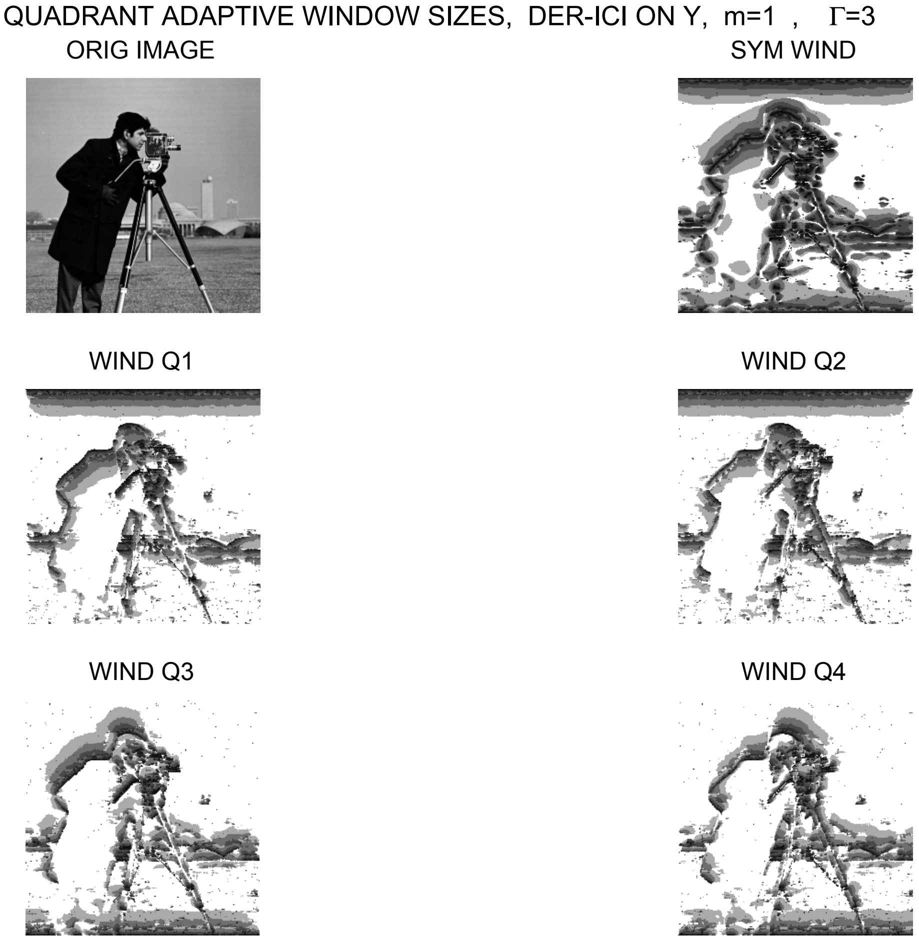


Figure 3.38: The *ICI* adaptive varying window sizes obtained for $\partial/\partial Y$ derivative estimates. The window sizes are presented for the symmetric and quadrant windows.

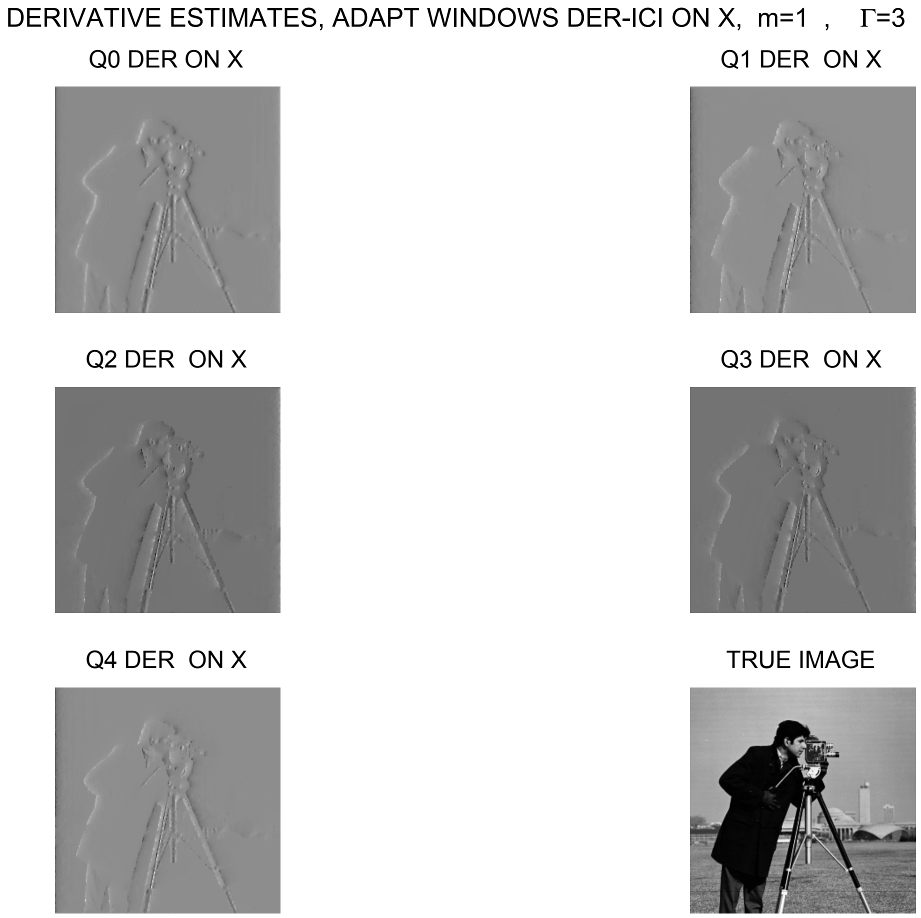


Figure 3.39: $\partial/\partial X$ derivative estimates with varying ICI adaptive window sizes obtained for the corresponding estimates $\partial/\partial X$.

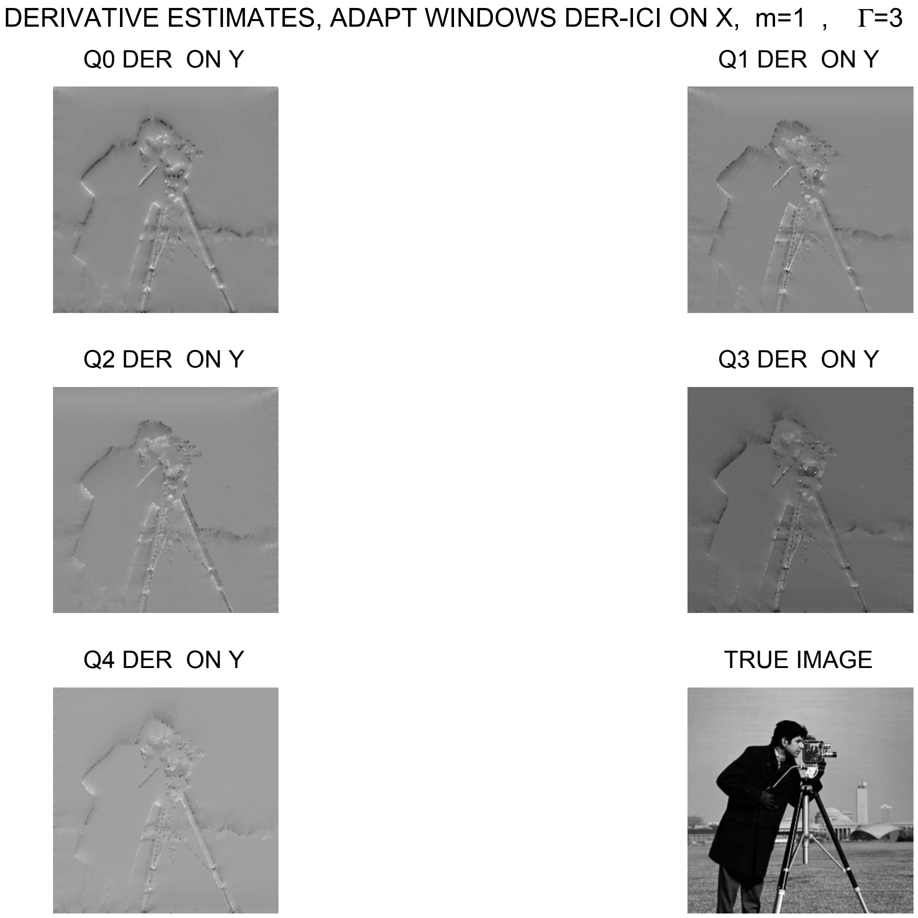


Figure 3.40: $\partial/\partial Y$ derivative estimates with varying ICI adaptive window sizes obtained for the estimates of the derivative $\partial/\partial X$.

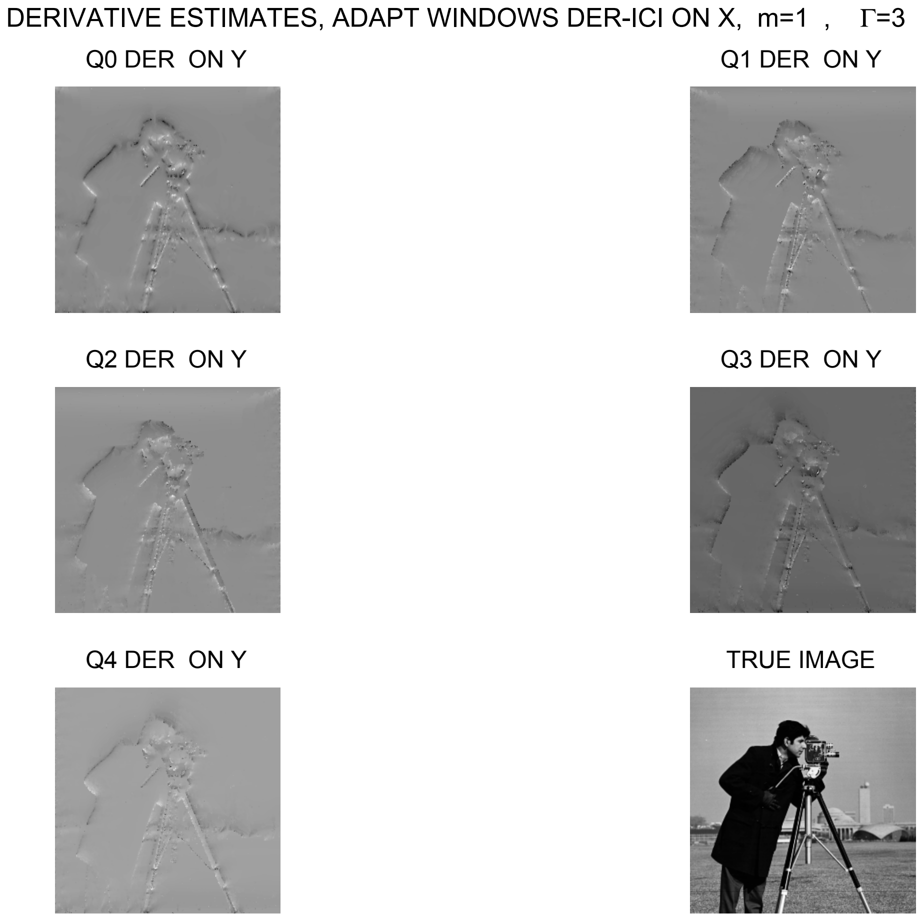


Figure 3.41: $\partial/\partial X$ derivative estimates with varying adaptive window sizes obtained for estimation of the derivative $\partial/\partial Y$.

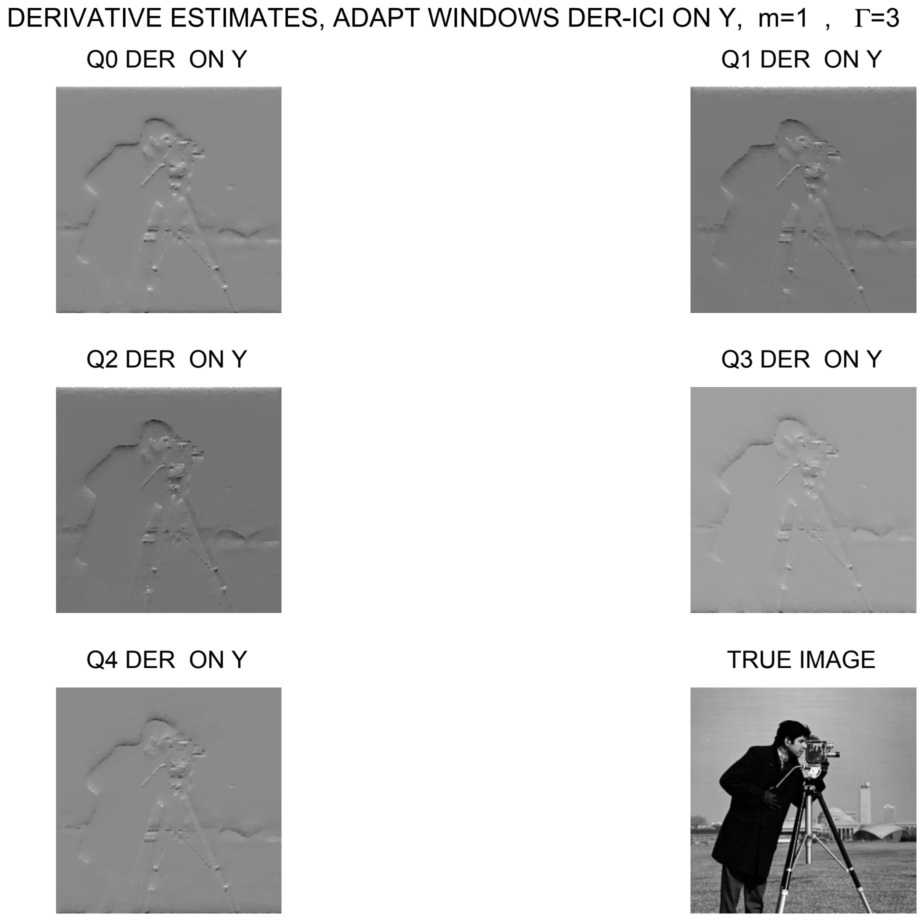


Figure 3.42: $\partial/\partial Y$ derivative estimate with the varying ICI adaptive window sizes obtained for estimation of the derivative $\partial/\partial Y$.

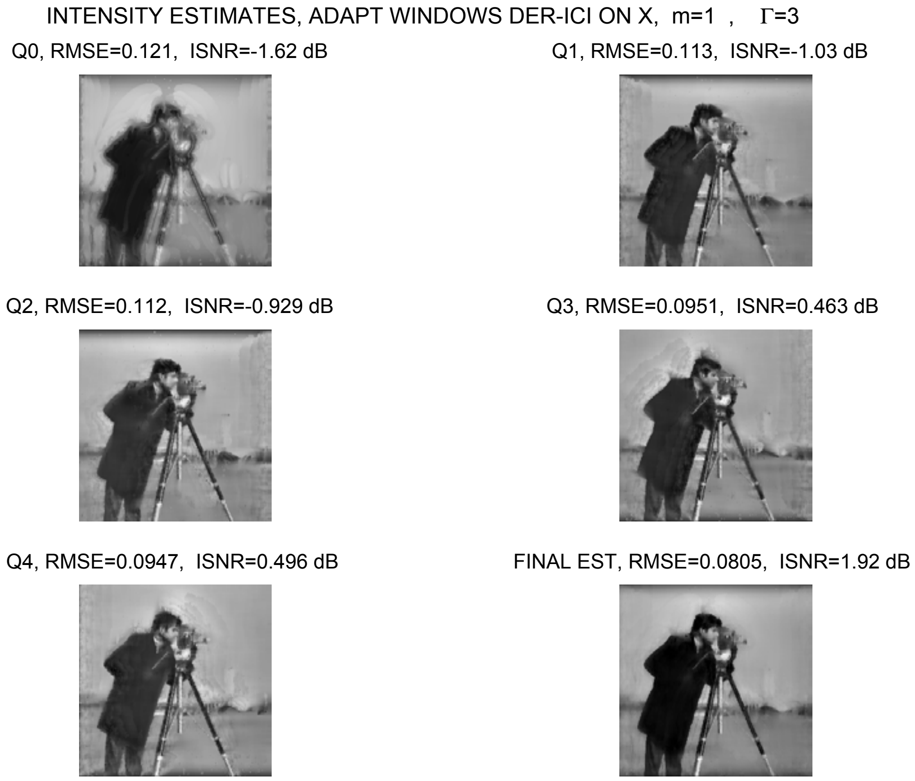
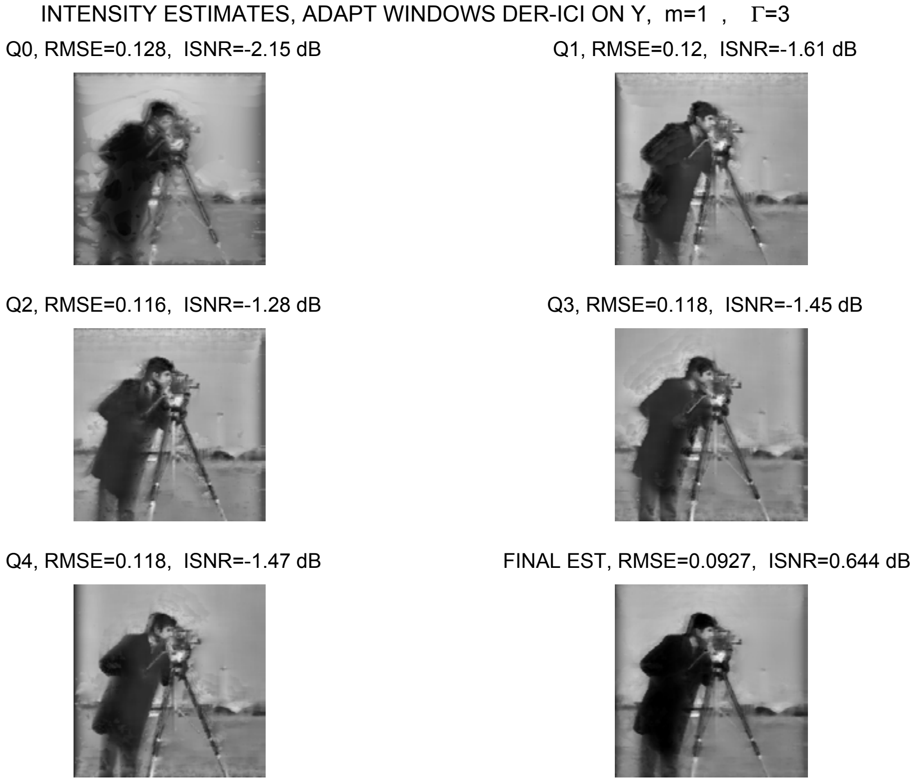

INTENSITY ESTIMATES, ADAPT WINDOWS DER-ICI ON X, m=1 , Γ=3
Q0, RMSE=0.121, ISNR=-1.62 dB
Q1, RMSE=0.113, ISNR=-1.03 dB
Q2, RMSE=0.112, ISNR=-0.929 dB
Q3, RMSE=0.0951, ISNR=0.463 dB
Q4, RMSE=0.0947, ISNR=0.496 dB
FINAL EST, RMSE=0.0805, ISNR=1.92 dB

Figure 3.43: The estimation of "Cameraman" with the varying adaptive window sizes obtained for estimation of the derivative $\partial/\partial X$.



INTENSITY ESTIMATES, ADAPT WINDOWS DER-ICI ON Y, m=1 , Γ=3

Q0, RMSE=0.128, ISNR=-2.15 dB

Q1, RMSE=0.12, ISNR=-1.61 dB

Q2, RMSE=0.116, ISNR=-1.28 dB

Q3, RMSE=0.118, ISNR=-1.45 dB

Q4, RMSE=0.118, ISNR=-1.47 dB

FINAL EST, RMSE=0.0927, ISNR=0.644 dB

Figure 3.44: Estimates of "Cameraman" with varying ICI adaptive window sizes obtained for estimation of the derivative $\partial/\partial Y$.

Chapter 4

ICI-ADAPTIVE MEDIAN FILTERING

4.1 Introduction

Median and weighted median filters have received considerable attention in signal processing over the last two decades. These structures are widely known in the signal processing literature as filters or smoothers. During last few years, the theory behind these median filters has been developing quite fast. Today, due to its sound underlying theory, weighted median smoothers are increasingly being used particularly in image processing applications. The success of median smoothers in image processing is based on two intrinsic properties: edge preservation and efficient attenuation of impulsive noise—properties not shared by traditional linear filters.

It is often stated that there are many analogies between weighted median smoothers and linear finite impulse response filters. While the median filters are definitely produce a nonlinear data transform, surprisingly, the analogy between the linear finite impulse response filters and nonlinear median filters becomes more evident if an input signal has an additive noise random component. In this case, the "sign" nonlinearity of the median is replaced by its smooth version with respect to the nonrandom component of the input signal. It is, so-called, a stochastic linearization effect (e.g. [103]).

Median filters are well known for being able to remove impulse noise and preserve image edges. When employed for image processing, however, median filters often exhibit blurring (oversmoothing) for large window sizes, or insufficient noise suppression for small window sizes. A number of an adaptive window-size median filter have been proposed, which can achieve a better performance of noise suppression and still preserve image sharpness. Most of this sort of algorithms are based on edge or impulse noise detection [35], [106], [36].

We describe a novel approach to solve a problem of varying scale selection for median filtering a noisy signal as it is proposed in [65], [70], [18]. The approach is based on the basic intention to achieve the best accuracy in signal denoising without specific algorithms included for edge detection and impulse noise removal.

We follow the idea of the ICI rule for the adaptive varying scale selection as it was presented above for the LPA linear estimates. It is shown that similar ideas with some technical modifications can be extended for the nonlinear median style estimates. It gives algorithms, which are simple to implement and nearly ideal in the asymptotic MSE accuracy.

The developed ICI rule for the median filters gives the adaptive varying window sizes and enables the algorithm to be spatially adaptive. Nonsymmetric and multiple window estimates combining the left, right and symmetric windowed median estimates are proposed.

In this chapter we start from the case of $1D$ signal and give the basic ideas and results, which enlighten the background of the proposed adaptive nonlinear estimates. The development for $2D$ image signal is demonstrated by simulation experiments.

4.2 Motivation

Suppose that we are given noisy observations of a signal $y(x)$ with a sampling period Δ :

$$z(x_s) = y(x_s) + \varepsilon(x_s),\ x_s = s\Delta,\ s = 1, 2, ...n, \tag{4.1}$$

where $\varepsilon(x_s)$ are independent and identically distributed random errors, $E(\varepsilon(x_s)) = 0$, $E(\varepsilon^2(x_s)) = \sigma^2$. It is not assumed that $\varepsilon(x_s)$ are Gaussian.

Let $y(x)$ belong to a nonparametric class of piece-wise smooth differentiable functions with a small number of discontinuities in the signal or its derivatives. The goal is to reconstruct $y(x_s)$ from the observations $(z(x_s))_{s=1}^n$ in such a way that certain desirable features such as jumps or instantaneous slope changes be preserved and the pointwise MSE risk be as small as possible.

As a basic estimator we study the median. It is well known that the sliding median estimate with respect to the additive Gaussian noise is nearly as good as the sliding linear average, while the median estimate demonstrates a good resistance to the random impulse noise modelled by heavy-tailed probability density functions. Another important feature of the median estimates is their ability to preserve edges and discontinuities in curves, which are usually smoothed by linear filtering. These properties as well as some advantages of implementation explain the great interest to this class of the median filters in signal processing (e.g. [3]).

In order to meet the aforementioned goal we introduce a median based filter, which is new in two important aspects:

(1) We consider in parallel the medians with symmetric and nonsymmetric left and right windows and determine the filter output as a combination of the outputs of these medians;

(2) The window size of the medians is varying and adaptive to unknown smoothness of the estimated signal. The *ICI* rule is developed for data-driven window size selection.

In general, the following criteria function $J_h(x)$ can be applied in order to obtain the usual weighted median sliding window filter as a solution of the optimization problem

$$\hat{y}_h(x) = \arg(\min_C \ J_h(x)), \tag{4.2}$$

$$J_h(x) = \sum_s w_h(x - x_s)|z(x_s) - C|, \tag{4.3}$$

where the window $w_h(x) = w(x/h)/h$ be a function satisfying the conventional properties: $w(x) \geq 0$, $w(0) = \max_x w(x)$, and $\int_{-\infty}^{\infty} w(u)du = 1$. Here the window w and the scale parameter h are applied exactly as it has been done in Chapter 1 for design of the linear *LPA* estimates. The only special difference between (4.3) and (1.8) is the nonquadratic loss function of residuals $z(x_s) - C$ used in (4.3). The scale parameter h determines observations and their weights used in the estimate.

For the symmetric, nonsymmetric right and nonsymmetric left rectangular windows:

$$w(u) = \begin{cases} 1,\ |u| \leq 1/2, 0 \text{ otherwise; symmetric window,} \\ 1,\ 0 \leq u \leq 1,\ 0 \text{ otherwise; left window,} \\ 1,\ -1 \leq u \leq 0,\ 0 \text{ otherwise; right window.} \end{cases} \tag{4.4}$$

The corresponding symmetric, left and right median estimates have a form

$$\hat{y}_h(x) = \begin{cases} median(z_{k-N_h}, ..., z_k, ..., z_{k+N_h}) \text{ ,symmetric median,} \\ median(z_{k-N_h}, ..., z_k), \text{ left median,} \\ median(z_k, ..., z_{k+N_h}), \text{ right median,} \end{cases} \tag{4.5}$$

where $N_h = \lfloor h/\Delta \rfloor$ is the integral part of h/Δ.

Let us present the algorithm for the general weighted median corresponding to the criteria function $J_h(x)$ (4.3).

The following calculations are efficient.

Let $z_{(r)}$ be ordered observations $z(x_s)$ from (4.1)

$$z_{(1)} \leq z_{(2)} \leq z_{(3)} \ldots . \tag{4.6}$$

Order the normalized window weights

$$v_s(x) = w(x - x_s) / \sum_{s'} w(x - x_{s'})$$

on s accordingly to (4.6) and denote these ordered variables as $v_{(r)}(x)$.

Consider the sum

$$S_q = \sum_{r=1}^{q} v_{(r)}(x) \tag{4.7}$$

and find the minimal q from the condition $S_q \geq 1/2$, i.e. $\hat{q} = \min\{q : S_q \geq 1/2\}$.

Then, the weighted median is given as

$$\hat{y}_h(x_k) = z_{(\hat{q})}. \tag{4.8}$$

These calculations are repeated for all x.

It is well known that window size selection is crucial in the efficiency of the nonlinear median estimators for noisy observations. When h is relatively small, the estimator $\hat{y}_h(x)$ gives a good approximation of a smooth $y(x)$ and the estimation error has a small bias, but then fewer data are used and $\hat{y}_h(x)$ is more variable and sensitive to the noise. The best choice of h involves in statistical terms a trade-off between the bias and variance, which depends on the sampling period, the standard deviation of the noise and the smoothness of the signal $y(x)$.

This chapter presents a modification and development of the adaptive varying window size algorithms discussed in Chapter 1 to the nonlinear median estimates.

Several examples

We consider several examples that compare the standard symmetric window median versus its nonsymmetric left and right counterparts. For the test we take the step signal

$$y(x) = \begin{cases} 0, \ 0 \leq x \leq 1/2, \\ 1, \ 1/2 \leq x \leq 1, \end{cases} \tag{4.9}$$

which is simplest to demonstrate the ability of the algorithm to estimate a jump-wise singularity in a signal. We show also how valuable is selection of the correct window size for the windowed estimate. The experiments with the *ICI* rule mainly present qualitative results demonstrating what sort of improvements can be expected from the adaptive varying window size median algorithm.

The observations are shown in Figure 4.1a. The other curves in Figure 4.1 illustrate median estimates with the symmetric window (4.4). Results of smoothing with different fixed size windows are shown in Figure 4.1b, c,

d. The corresponding $RMSE$ values are given in the figures. Figure 4.1d presents the estimate with the ideal window size parameter, equal to $N_h = 29$, minimizing the mean-square error $\frac{1}{n}\sum_k (y(x_k) - \hat{y}_h(x_k))^2$, calculated provided that the true signal (4.9) is known. Smaller window with $N_h = 3$ (Figure 4.1b) preserves the jump in the signal better but has a larger random component in the estimate, while the larger window with $N_h = 87$ (Figure 4.1c) oversmooths the observations. Here we wish to emphasize that, as a matter of fact, for noisy data the ability of the conventional median filter to preserve edges and jumps in a signal is quite a questionable property. The median filter is not able to delineate the jump accurately and smooths the true signal.

It can be seen that this smoothing becomes stronger if the standard deviation of the noise increases. For comparison, in Figure 4.1e, we show the estimate given by the symmetric median filter equipped with the ICI rule used for the varying adaptive window size selection. The corresponding $RMSE = 0.0274$ is nearly the ideal $RMSE = 0.0273$ and in the contrast to the ideal results given in Figure 4.1d it is obtained assuming that the true signal is unknown. The corresponding varying values of the scale parameter h, adaptive according to the ICI rule, are shown in Figure 4.1f.

The curves of Figure 4.2a, c provide the adaptive varying values of h obtained for the left and right windows by the ICI rule. We wish to emphasize that the results given by these curves are in the accurate agreement with the window sizes that could be selected provided that we knew in advance that the signal is piece-wise constant with a single jump and the location of this jump is known exactly. Figure 4.2a shows the adaptive h for the left median estimate. In the first part of this curve, $0 \leq x \leq 1/2$, h is a linear increasing function of x. This happens because the left median uses for estimation only the observations, which are on the left-hand side from the estimation point x and all these observations are used for estimation of the constant signal value $y = 0$. Immediately after $x = 1/2$ the value of the window size drops as the signal value is changed to $y = 1$ and only a small number of the observations can be used for this estimates for x close to $1/2$. While x is increasing towards the value 1 the window size h is also increasing as a larger number of observations can be used for estimation of $y = 1$. Steps seen in the window size curve are caused by using a exponential grid for h in the ICI rule.

The left median estimate given into Figure 4.2b corresponds to the window sizes presented in Figure 4.2a. We can see that the estimate delineates accurately the jump in the signal from its left-hand side perfectly suppressing the noise. Naturally the picture is not so perfect on the right-hand side of the jump because only small number of the observations can be used by the left median for estimation for values of x, which are nearly the point $x = 1/2$.

Similar interpretation can be given for the curves in Figures 4.2c, d shown for the adaptive window size and estimates by the right median estimator. Figures 4.2e, f show the combined estimates obtained by fusing, respectively, two left and right medians, and three left, right and symmetric medians. The combined left-right estimate (Figure 4.2e) gives a nearly perfect reconstruction of the jump, while the estimate combining the three estimates (Figure 4.2f) is a bit worse in the area of the jump. A comparison of these combined estimates using the non-symmetric estimates shows a clear advantage with respect to the symmetric median estimates shown in Figure 4.1.

In Figure 4.3 we show the estimates obtained by using the left and right medians provided that the window size is invariant and equal to the window sizes used for the symmetric window estimates in Figure 4.1. The true signal is shown by the dotted line. It is obvious from the curves that these nonsymmetric median estimates shift the point of the jump and this shift is quite large for the larger h. The important conclusion follows that the left and right median estimates are not able to locate a jump without the varying adaptive window size.

In order to achieve an improvement we should use both the nonsymmetric estimators and the varying window size selector.

Concerning the accuracy of estimation we wish to note that the ICI rule has a design parameter Γ, which actually controls the smoothness of the adaptive window size given by the ICI rule. In general, larger values of Γ tend to result in a smoother window size h as a function of x, while smaller values of Γ result in a larger variability in h. We use in Figures 4.2a quite large values $\Gamma = 3$ in order to expose a basic tendencies in the varying adaptive window size. The $RMSE$ accuracy improvement usually requires smaller values of Γ. Then, the adaptive window size h as a function of x is very irregular and noisy. Figures 4.4a,c show the curves of the adaptive left and right ICI window sizes for $\Gamma = 1.2$, while the h for the symmetric window is given in Figure 4.1f.

The $RMSE$ values shown in Figure 4.1 and Figure 4.4 make possible a more accurate quantitative comparison of the methods. Firstly note, that the left and right median estimates in Figures 4.4b, d yield the accuracy, which is nearly optimal for the symmetric median. The fused estimates shown in Figures 4.4e,f give $RMSE$ values, which are as much as twice better than that for the symmetric ideal median estimate in Figure 4.1d.

These experiments show that we can expect quite a valuable accuracy improvement from the adaptive methods as well as a more accurate reconstruction of curves with discontinuities.

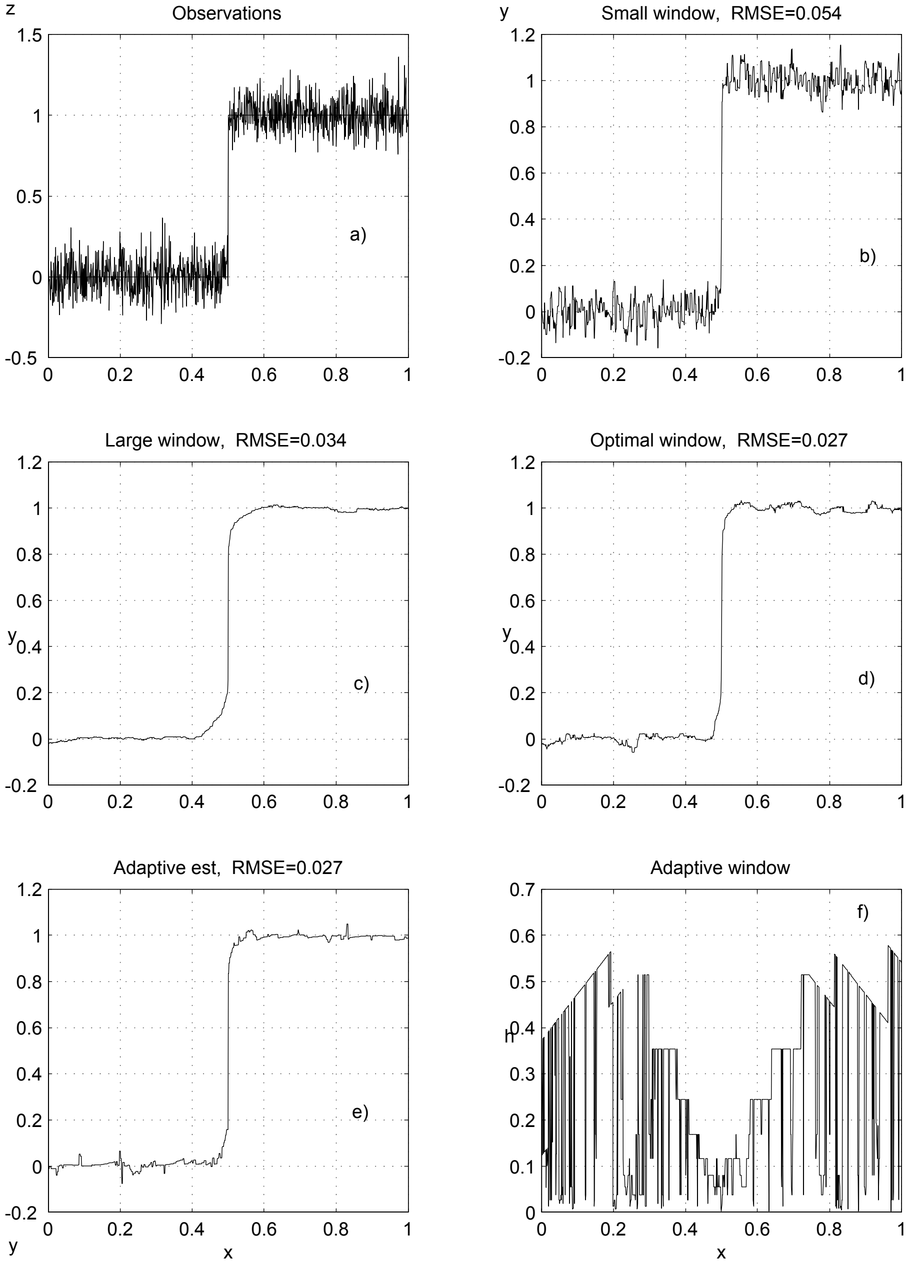


Figure 4.1: Symmetric window median estimates of the step-wise signal: a) Observations, $N = 1024$, $\sigma = 0.1$; b),c),d) Estimates with invariant window size: $N_h = 3$, 87, 29, respectively; e) Estimate with the ICI adaptive window size; f) Adaptive window sizes obtained by the ICI rule.

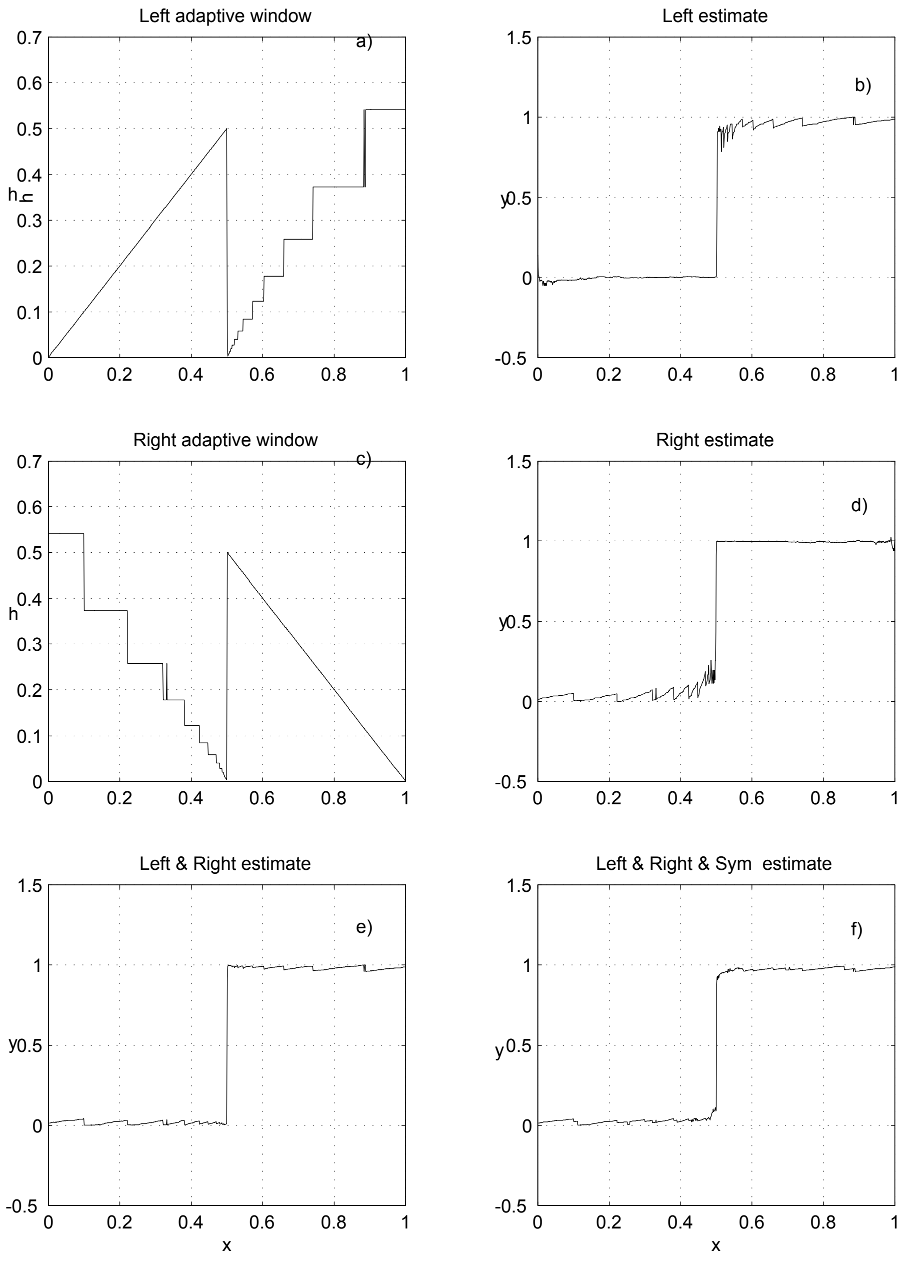


Figure 4.2: Adaptive nonsymmetric left and right window median estimates of the step signal: a), b) Window size and estimate given by the left window median; c), d) Window size and estimate given by the right window median; e) L & R estimate fusing the left and right adaptive median estimates; f) L & R & S estimate fusing the left, right and symmetric adaptive median estimates. Threshold parameter $\Gamma = 3$.

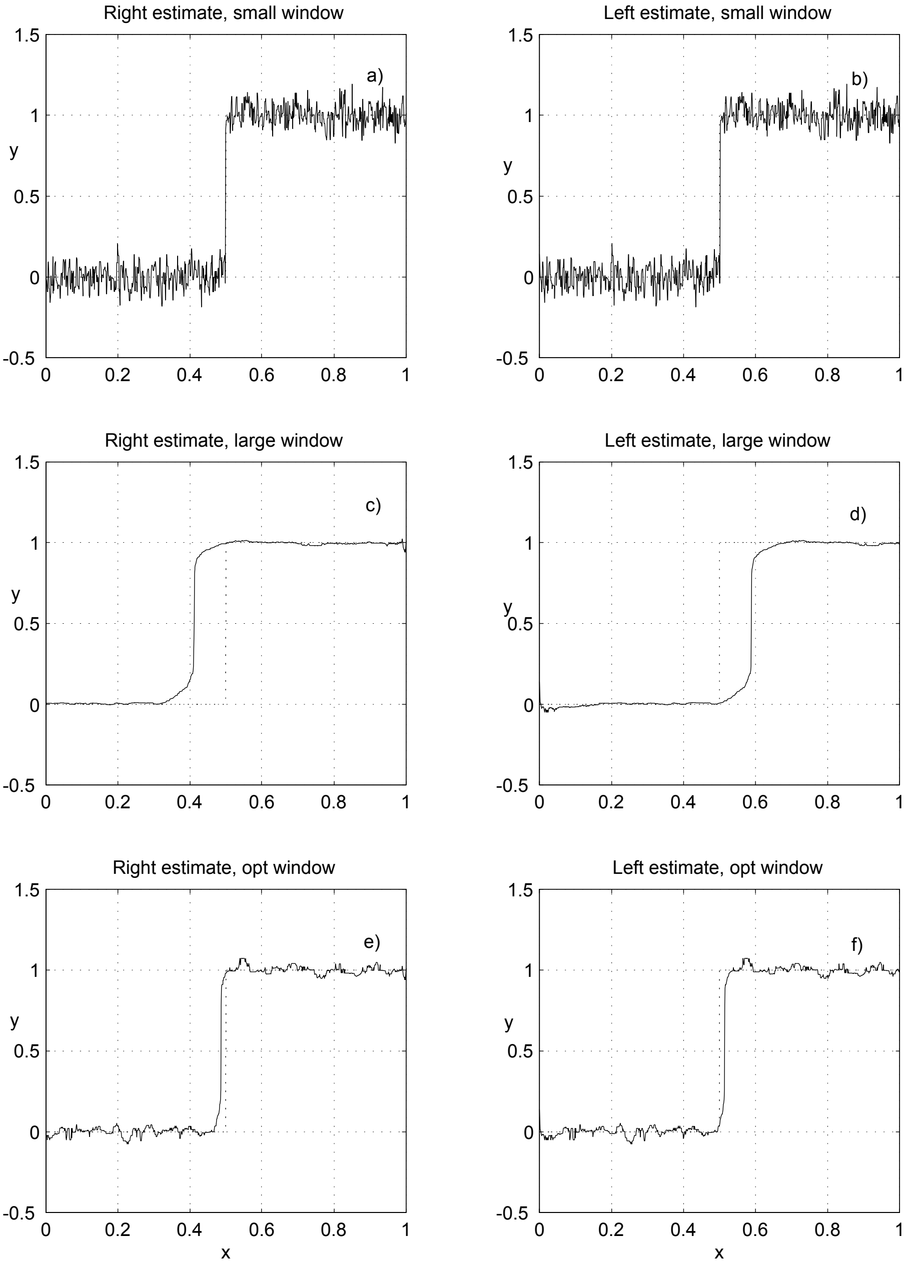


Figure 4.3: Nonsymmetric left and right window median estimates with invariant window size given for the observations presented in Figure 1a: a), b) $N_h = 3$; c), d) $N_h = 87$; e), f) $N_h = 29$.

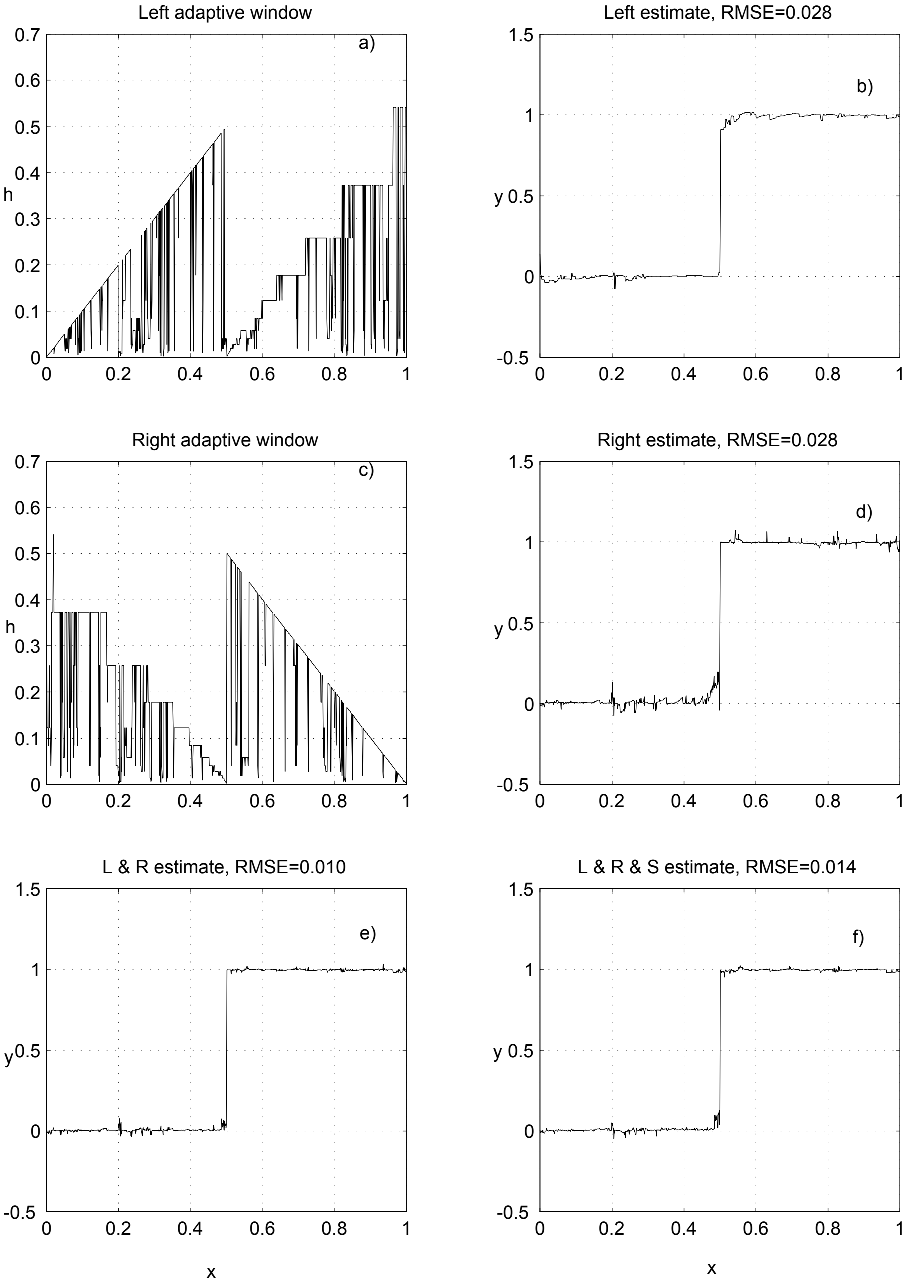


Figure 4.4: Adaptive nonsymmetric left, right window and combined median estimates of the step-wise signal: a), b) Window size and estimate given by the left window median; c), d) Window size and estimate given by the right window median; e) L & R estimate fusing the left and right adaptive median estimates; f) L & R & S estimate fusing the left, right and symmetric adaptive median estimates. Threshold parameter $\Gamma = 2.0$ corresponds to improved estimation accuracy.

4.3 $1D$ algorithm

4.3.1 ICI rule for window size selection

What follows in this section is quite similar to the accuracy analysis and the ICI rule introduction in Chapter 1 given for the linear estimates.

Let $e(x,h) = y(x) - \hat{y}_h(x)$ be the error of the median estimate given by (4.2).

It follows from Proposition 4 and formulas (4.59) and (4.60) presented latter in this chapter that asymptotically for $\Delta \to 0$, $h \to 0$, $h/\Delta \to \infty$ and the Gaussian noise the standard deviation $\sigma(x,h)$ of the estimate error has a form:

$$\sigma(x,h) = \sigma\sqrt{\frac{\Delta\pi}{2h}\int w^2(u)du}, \tag{4.10}$$

where σ is the standard deviation of the additive Gaussian noise in the observation model,

and

$$m_e(x,h) \leq \gamma \cdot \sigma(x,h) \text{ if } h \leq h^*(x), \tag{4.11}$$

where $m_e(x,h)$ is the estimation bias and $h^*(x)$ is the ideal window size giving the minimum value of the MSE provided that the signal is known.

Then it can be written for the estimation error:

$$|e(x,h)| = |y(x) - \hat{y}_h(x)| \leq m_e(x,h) + |e^0(x,h)|,$$

where asymptotically the term $e^0(x,h)$ is Gaussian.

With the probability $p = 1 - \alpha$ the following inequality holds

$$|e(x,h)| \leq m_e(x,h) + \chi_{1-\alpha/2}\sigma(x,h), \tag{4.12}$$

where $\chi_{1-\alpha/2}$ is $(1-\alpha/2)-th$ quantile of the standard Gaussian distribution.

Introduce a finite set of increasing window sizes:

$$H = \{h_1 < h_2 < .... < h_J\},$$

starting from quite a small h_1.

Then, according to (4.11) the inequality (4.12) can be weakened for $h \leq h^*(x)$ to

$$|e(x,h)| \leq \Gamma \cdot \sigma(x,h), \tag{4.13}$$
$$\Gamma = \gamma + \chi_{1-\alpha/2}. \tag{4.14}$$

We use the inequalities (4.13) corresponding to different h in order to test the hypothesis $h \leq h^*(x)$ and to find the value of h close to $h^*(x)$.

According to (4.13) determine a sequence of the confidence intervals $D(j)$

of the biased estimates as follows

$$D(j) = [\hat{y}_{h_j}(x) - \Gamma \cdot \sigma(x, h_j),\ \hat{y}_{h_j}(x) + \Gamma \cdot \sigma(x, h_j)], \tag{4.15}$$

where Γ is the threshold of the confidence interval.

Then for $h = h_j$ the inequality (4.13) is of the form

$$y(x) \in D(j), \tag{4.16}$$

and we can conclude from (4.12) that as long as the inequality $h \leq h^*(x)$ holds for $h = h_j$, $1 \leq j \leq i$, all the intervals $D(j)$, $1 \leq j \leq i$, have a point in common, namely, $y(x)$.

Now the ICI rule can be formulated in the form indistinguishable from the one given for the linear estimate case (2.7).

Consider the intersection of the intervals $D(j)$, $1 \leq j \leq i$, with increasing i, and let i^+ be the largest of those i for which the intervals $D(j)$, $1 \leq j \leq i$, have a point in common. This i^+ defines the adaptive window size and the adaptive median estimate as follows

$$\hat{y}^+(x) = \hat{y}_{h^+(x)}(x),\ h^+(x) = h_{i^+}. \tag{4.17}$$

The following algorithm implements the procedure (4.17). Determine the sequence of the upper and lower bounds of the confidence intervals $D(i)$ as follows (2.8)

$$\begin{aligned} D(i) &= [L_i,\ U_i], \\ U_i &= \hat{y}_{h_i}(x) + \Gamma \cdot \sigma(x, h_i), \\ L_i &= \hat{y}_{h_i}(x) - \Gamma \cdot \sigma(x, h_i). \end{aligned} \tag{4.18}$$

Let

$$\begin{aligned} &\bar{L}_{i+1} = \max[\bar{L}_i,\ L_{i+1}],\ \underline{U}_{i+1} = \min[\underline{U}_i,\ U_{i+1}], \\ &i = 1, 2, ..., J,\ \bar{L}_1 = L_1,\ \underline{U}_1 = U_1 \end{aligned} \tag{4.19}$$

then the optimal window length h_i^+ comes for the largest i, i^+, for which the inequality

$$\bar{L}_i \leq \underline{U}_i \tag{4.20}$$

is still satisfied. This i^+ is the largest of those i for which the confidence intervals $D(i)$ have a point in common as discussed above. This ICI window size selection procedure requires knowledge of the estimate $\hat{y}_{h_i}(x)$ and its standard deviation $\sigma(x, h_i)$ only. The procedure described above is repeated for the every x.

The adaptive estimator is implemented as J parallel filters, which are different only by the window sizes h_j, $j = 1, 2, ...J$, and the selector, which determine the best $h^{+}(x)$ and the corresponding estimate $\hat{y}_{h^{+}(x)}(x)$ for the every x. The selector uses the rule based on the ICI statistic. The adaptive algorithm is nearly ideal within $\ln n$ factor in the point-wise risk for estimating the signal.

4.3.2 Multiple window estimates

(1) Let w_L, w_R and w_S be the left, right, and symmetric windows and $\hat{y}_L$, $\hat{y}_R$, and $\hat{y}_S$ be the corresponding estimates of $y(x)$.

Then the combined median estimate $\hat{y}$ can be produced in the form of the three-

$$\begin{aligned} &\hat{y}_{LRS} = \lambda_L \hat{y}_L + \lambda_R \hat{y}_R + \lambda_S \hat{y}_S, \\ &\lambda_L = \frac{\sigma_L^{-2}}{\sigma^{-2}},\ \lambda_R = \frac{\sigma_R^{-2}}{\sigma^{-2}},\ \lambda_S = \frac{\sigma_S^{-2}}{\sigma^{-2}}, \\ &\sigma^{-2} = \sigma_R^{-2} + \sigma_L^{-2} + \sigma_S^{-2}, \end{aligned} \tag{4.21}$$

or two fused estimates

$$\begin{aligned} &\hat{y}_{LR} = \lambda_L \hat{y}_L + \lambda_R \hat{y}_R, \\ &\lambda_L = \frac{\sigma_L^{-2}}{\sigma^{-2}},\ \lambda_R = \frac{\sigma_R^{-2}}{\sigma^{-2}}, \\ &\sigma^{-2} = \sigma_R^{-2} + \sigma_L^{-2}, \end{aligned} \tag{4.22}$$

with the inverse standard deviations used as weights and the adaptive varying windows selected independently for the left-, right- and symmetric window estimators.

(2) Let $h_L^{+}(x_k)$ and $h_R^{+}(x_k)$ be the left and right adaptive rectangular window sizes obtained by the ICI rule for $x = x_k$ and let $N_L^{+}(x_k) = \lfloor h_L^{+}(x_k)/\Delta \rfloor$ and $N_R^{+}(x_k) = \lfloor h_R^{+}(x_k)/\Delta \rfloor$ be numbers of the observations corresponding to the left and right adaptive window sizes.

Let us exploit these adaptive window sizes in order to introduce the adaptive weighted median estimate with the combined window as it was discussed in Section 2.7. In general, this estimate has a nonsymmetric window with the sizes of the left and right parts of the window given by the values $N_L^{+}(x_k)$ and $N_R^{+}(x_k)$.

This combined window weighted median estimate is defined as a solution of the problem

$$\begin{aligned} &\hat{y}_{LRW}(x_k) = \arg\min_{C} J(x_k) \\ &J(x_k) = \sum_{s=1}^{N_R^{+}(x_k)} W_R(s)|z_{k+s} - C| + \sum_{s=1}^{N_L^{+}(x_k)} W_L(s)|z_{k-s} - C| + \\ &|z_k - C| W_M, \end{aligned} \tag{4.23}$$

where $W_R(k)$, $W_L(k)$ and W_M are the weights of the right, left and middle observations respectively.

The values of this weights depends on $N_L^+(x_k)$ and $N_R^+(x_k)$. In particular, for the triangular window:

$$\begin{aligned} W_M &= \max(N_L^+(x_k), N_R^+(x_k)), \\ W_R(s) &= W_M - s,\ \ s = 1, ..., N_R^+(x_k), \\ W_R(s) &= W_M - s,\ \ s = 1, ..., N_L^+(x_k). \end{aligned} \tag{4.24}$$

The combined window left-right weighted median estimate $\hat{y}_{LRW}$ (4.23) is different from the one given by the formulas (4.21) and (4.22). The estimates (4.21) and (4.22) are obtained by fusing the left, right and symmetric adaptive estimates. In (4.23)-(4.24) we apply the ICI rule only in order to find the window sizes and the weighted median estimate is calculated independently from the ICI procedure using this adaptive non-symmetric window.

4.3.3 Adjustment of threshold

The threshold parameter Γ in (4.15) plays an important role in the performance of the algorithm. Too large or too small Γ results in oversmoothing or undersmoothing data, respectively. Speculations and methods used for median estimates are completely similar to ones discussed for the $ICI - LPA$ estimates in Section 2.4. Here we wish consider the use of the CV criteria and its specification for the weighted median estimates.

The direct implementation of the one-out CV approach assumes a calculation of the criteria function

$$I_{CV} = \sum_k (z(x_k) - \hat{y}^{[k]}_{h^+(x_k)}(x_k))^2, \tag{4.25}$$

where the index $[k]$ indicates that the estimate for the point x_k is calculated provided that the observation $z(x_k)$ is omitted. Thus, the calculation of I_{CV} assumes permutations of the observations, that forms special sets of the observations for the every x_k. As a matter of fact the criteria function I_{CV} serves as an evaluation test of predictive properties of the estimates. Amount of calculations required by the permutational criteria function (4.25) is prohibitive for many cases.

It is well known that for the linear estimates the criteria I_{CV} (4.25) can be calculated in one pass without the permutations using the estimates obtained on whole set of the observations only. Special weights of the estimation residuals are used in these algorithms (e.g. [60]).

Motivated by this idea, let us first consider the linear estimates obtained from (4.2), where the module is replaced by the second power of the residuals. Then, the corresponding linear estimate has a form (1.31)

$$\hat{y}_h(x) = \sum_s g_h(x - x_s) z(x_s),$$
$$g_h(x - x_s) = \frac{w_h(x - x_s)}{\sum_s w_h(x - x_s)}$$

and according to (2.12) the CV criteria is as follows

$$\hat{I}_{CV} = \sum_k \left(\frac{z(x_k) - \hat{y}_h(x_k)}{1 - g_h(0)} \right)^2, \tag{4.26}$$
$$g_h(0) = \frac{w_h(0)}{\sum_s w_h(x - x_s)}.$$

For the ICI adaptive window size $h = h^+(x_k)$ these formulas are transformed to

$$\hat{I}_{CV} = \sum_k \left(\frac{z(x_k) - \hat{y}_{h^+(x_k)}(x_k)}{1 - g_{h^+(x_k)}(0)} \right)^2,$$
$$g_{h^+(x_k)}(0) = \frac{w_{h^+(x_k)}(0)}{\sum_s w_{h^+(x_k)}(x - x_s)}.$$

For the rectangular window w it gives

$$\hat{I}_{CV} = \sum_k \left(\frac{z(x_k) - \hat{y}_{h^+(x_k)}(x_k)}{1 - 1/n^*(x_k)} \right)^2, \tag{4.27}$$
$$g_{h^+(x_k)}(0) = 1/n^+(x_k),$$

where $n^+(x_k)$ is a number of observations used in the optimal window $h^+(x_k)$.

It means that

$$n^+ = \begin{cases} N_h(x_k), & \text{for the nonsymmetric window,} \\ 2N_h(x_k) - 1, & \text{for the symmetric window.} \end{cases} \tag{4.28}$$

The *generalized* CV following from (4.27) according to some general ideas of the CV [122] is of the form:

$$\hat{I}_{GCV} = \frac{\sum_k (z(x_k) - \hat{y}_{h^+(x_k)}(x_k))^2}{\sum_k (1 - 1/n^+(x_k))^2}. \tag{4.29}$$

It is emphasized that the CV criteria (4.27) and (4.29) are derived and justified for the linear estimates only.

Now let us go back to the nonlinear median estimates. Remind that the estimate $\hat{y}_h(x)$ following from minimization of (4.1) is a solution of the equation

$$\sum_s w_h(x - x_s) sign(z(x_s) - C) = 0. \tag{4.30}$$

Let us use the so-called stochastic linearization of nonlinear functions (e.g. [103]). Then, the nonlinear *sign* function can be approximated by some linear function as follows

$$sign(r_s) \simeq \alpha_s r_s, \; r_s = z(x_s) - C. \tag{4.31}$$

According to the stochastic linearization technique the coefficient α_s is defined by minimizing the mean squared error of approximation, i.e.

$$\alpha_s = \arg\min_{\alpha} E\{(sign(r_s) - \alpha_s r_s)^2\}.$$

Simple transformations give

$$\alpha_s = \frac{E\{r_s \cdot sign(r_s)\}}{\sigma_s^2}, \; \sigma_s^2 = E\{(r_s)^2\}.$$

Assume that the residuals e_s are Gaussian $N(0, \sigma_s)$, then it can be found that

$$\alpha_s = \sqrt{\frac{2}{\pi}} \frac{1}{\sigma_s}. \tag{4.32}$$

Substituting (4.32) and (4.31) into (4.30) gives

$$C = \frac{\sum_s \frac{1}{\sigma_s} w_h(x - x_s) z(x_s)}{\sum_s \frac{1}{\sigma_s} w_h(x - x_s)}.$$

Thus, we arrive to the approximate median estimate in the linearized form

$$\hat{y}_h(x) \simeq \sum_s g_h(x - x_s) z(x_s), \tag{4.33}$$

$$g_h(x - x_s) = \frac{\frac{1}{\sigma_s} w_h(x - x_s)}{\sum_s \frac{1}{\sigma_s} w_h(x - x_s)}$$

where the standard deviation σ_s depends on the estimate as $\sigma_s^2 = E\{(z(x_s) - \hat{y}_h(x_s))^2\}$.

Actually, nonlinear equations can be derived for σ_s^2 as it is usually in the stochastic linearization technique. However, we are not going to develop this line further as our goal only to show that to some extend the nonlinear median estimate can be treated as an approximated linear one.

Further, let us assume for the simplicity that all of σ_s are more less equal to each other. Then

$$g_h(x - x_s) \simeq \frac{w_h(x - x_s)}{\sum_s w_h(x - x_s)}, \tag{4.34}$$

which approximately coincides with given in (4.26).

Then, we are able to apply the CV criteria functions (4.26) and (4.29) for the weighted median estimates. In this way, we show that the proposed CV criteria can be reasonably used for the nonlinear median estimates.

The assumptions concerning the Gaussian nature of the estimates as well as the equality of the standard deviations σ_s can be justified only in terms of the asymptotic consideration with a large number of observations produced with a small sampling period. Actually the idea of using the CV criteria function is applicable for the median estimates (left, right and symmetric) in the form (4.27)-(4.29) and quite questionable for the fused and combined window weighted estimates. Some simulation experiments confirm these speculations.

We applied (4.27)-(4.29) as a criteria function for a data-driven selection of the threshold parameter Γ. It assumes that the procedure (4.18)-(4.20) to be repeated for every $\Gamma \in G$, $G = \{\Gamma_1, \Gamma_2, ...\Gamma_{N_G}\}$, and

$$\hat{\Gamma} = \arg\min_{\Gamma \in G} \hat{I}_{GCV} \tag{4.35}$$

gives the adjusted threshold parameter value.

The criteria (4.29) was used also for the combined estimates (4.21) and (4.22) provided that the estimates $\hat{y}_{LRS}$ or $\hat{y}_{LR}$ are used instead of $\hat{y}_{h^+(x_k)}(x)$ in the CV criteria, where $n^+(x_k)$ means a total number of observations entered in these estimates with nonzero weights.

4.4 Performance analysis for $1D$ signal

4.4.1 Asymptotic accuracy with fixed scale

For the accuracy analysis we will consider a class of estimates more general than the weighted median. Let the estimate of the signal is defined as a solution of the following optimization problem:

$$\hat{y}_h(x) = \arg(\min_C J_h), \tag{4.36}$$
$$J_h(x) = \sum_s w_h(x - x_s)F(z(x_s) - C),$$

where, in general, the loss function $F(x)$ is convex, bounded and even, $F(x) = F(-x)$.

In statistics, the term M-estimate is usually used for estimates obtained by minimizing a sum of nonquadratic loss functions of residuals. Thus, the $\hat{y}_h(x)$ given by (4.36) is the M-estimate of the signal and the median is a special case of this class of estimates with $F(x) = |x|$.

We use the following assumptions concerning the loss function F and the distribution of the random observation errors ε, which can be non-Gaussian:

1^0. The random $\varepsilon(x_s)$ are independent and identically distributed for all x_s with the symmetric distribution

$$P_\varepsilon(x) = 1 - P_\varepsilon(-x). \tag{4.37}$$

For the probability density it means

$$p_\varepsilon(x) = p_\varepsilon(-x),\ p_\varepsilon(x) = \partial P_\varepsilon(x)/\partial x;$$

2^0. The loss function $F(x)$ is convex, bounded, twice differentiable almost everywhere and symmetric

$$F(x) = F(-x); \tag{4.38}$$

3^0. The following holds

$$E\{F^{(1)}\} \triangleq \int F^{(1)}(v)dP_\varepsilon(v) = 0, \tag{4.39}$$

$$0 < E\{(F^{(1)})^2\} \triangleq \int (F^{(1)}(v))^2 dP_\varepsilon(v) < \infty, \tag{4.40}$$

$$0 < E\{F^{(2)}\} \triangleq \int F^{(2)}(v)dP_\varepsilon(v) < \infty, \tag{4.41}$$

where $F^{(k)}(x) = d^k F(x)/dx^k$, $k = 1, 2$.

4^0. Denote

$$V(F,P_\varepsilon)=\frac{E\{(F^{(1)})^2\}}{(E\{F^{(2)}\})^2}. \tag{4.42}$$

Actually, the zero mean value of $F^{(1)}(x)$ follows from (4.37) and (4.38).

It is assumed in the considered asymptotic that h and Δ are as follows :

$$\Delta\rightarrow 0,\ h\rightarrow 0,\ h/\Delta\rightarrow\infty. \tag{4.43}$$

The last assumption $h/\Delta\rightarrow\infty$ means that a number of observations in the window w_h is increasing. Actually, these assumptions are used in order to have small estimation errors:

$$e(x,h)=y(x)-\hat{y}_h(x)\rightarrow 0.$$

The necessary minimum condition in (4.36) has a form

$$\begin{aligned}&\sum_s w_h(u_s)F^{(1)}(r_s)=0, \qquad (4.44)\\ &r_s=y(x-u_s)+\varepsilon(x_s)-\hat{y}_h(x),\\ &u_s=x-x_s,\end{aligned}$$

where r_s is a residual of estimation.

The polynomial expansion of $y(x-u_s)$ on small u_s gives for the residuals:

$$\begin{aligned}&r_s=y(x)-y^{(1)}(x)u_s+y^{(2)}(x)u_s^2/2+\varepsilon(x_s)-\hat{y}_h(x)=\\ &e(x,h)-y^{(1)}(x)u_s+y^{(2)}(x)u_s^2/2+\varepsilon(x_s),\end{aligned}$$

and it results for (4.44) in the equation

$$\sum_s w_h(u_s)\left[F^{(1)}(\varepsilon(x_s))+F^{(2)}(\varepsilon(x_s))(e(x,h)-y^{(1)}(x)u_s+y^{(2)}(x)u_s^2/2)\right]=0. \tag{4.45}$$

It can be shown that the random $\sum_s w_h(u_s)F^{(2)}(\varepsilon(x_s))$ converges in probability to its expectation. Asymptotically it gives

$$\begin{aligned}&\sum_s w_h(u_s)F^{(2)}(\varepsilon(x_s))\xrightarrow{P}\sum_s w_h(u_s)E_\varepsilon\{F^{(2)}(\varepsilon(x_s))\}\rightarrow \qquad (4.46)\\ &\frac{1}{\Delta h}\int w(u/h)du\cdot\overline{F^{(2)}}=\frac{1}{\Delta}\int w(u)du\cdot\overline{F^{(2)}}=\frac{1}{\Delta}\overline{F^{(2)}},\\ &\overline{F^{(2)}}=E_\varepsilon(F^{(2)}(\varepsilon)). \qquad (4.47)\end{aligned}$$

It is used in the above transformations, that $\int w(u)du=1$ and the mathematical expectation is calculated on the random observation errors ε.

In a similar way, we obtain

$$\sum_s w_h(u_s)u_s F^{(2)}(\varepsilon(x_s)) \xrightarrow{P} \frac{h}{\Delta}\int w(u)udu \cdot \overline{F^{(2)}}, \tag{4.48}$$
$$\sum_s w_h(u_s)u_s^2 F^{(2)}(\varepsilon(x_s)) \xrightarrow{P} \frac{h^2}{\Delta}\int w(u)u^2du \cdot \overline{F^{(2)}}.$$

Substituting (4.46) and (4.48) in (4.45) we obtain the equation

$$\sum_s w_h(u_s)F^{(1)}(\varepsilon(x_s)) + e(x,h)\frac{1}{\Delta}F^{(2)} - y^{(1)}(x)\frac{h}{\Delta}\int w(u)udu \cdot \overline{F^{(2)}} +$$
$$y^{(2)}(x)\frac{h^2}{\Delta}\int w(u)u^2du \cdot \overline{F^{(2)}} = 0,$$

which, being solved with respect to $e(x,h)$, gives

$$e(x,h) = m_e(x,h) + \xi(x,h), \tag{4.49}$$
$$m_e(x,h) = y^{(1)}(x)h\int w(u)udu - y^{(2)}(x)\frac{h^2}{2}\int w(u)u^2du,$$
$$\xi(x,h) = -\frac{\sum_s w_h(u_s)F^{(1)}(\varepsilon(x_s))}{\frac{1}{\Delta}F^{(2)}},$$

where $m_e(x,h)$ and $\xi(x,h)$ are the bias and random component of the estimation error.

As $E\{F^{(1)}\} = \overline{F^{(1)}} = 0$, according to (4.39), we have for the random $\xi(x,h)$

$$E\xi(x,h) = 0 \text{ and } E\{\xi^2(x,h)\} = V\frac{\Delta}{h}\int w^2(u)du, \tag{4.50}$$

where V is defined in (4.42).

Thus, we arrive at the following proposition.

Proposition 4 *Let the assumptions (4.37) - (4.43) hold. Then, the estimation error of the filter (4.36) can be represented in the form*

$$e(x,h) \simeq m_e(x,h) + \xi(x,h), \tag{4.51}$$

where

$$E\xi(x,h) \simeq 0, \ var(\xi(x,h)) \simeq V\frac{\Delta}{h}\int w^2(u)du,$$

and for the bias of estimation we have

$$m_e(x,h) \simeq \begin{cases} -\frac{1}{2}h^2y^{(2)}(x)\int w(u)u^2du, \ \text{if } \int w(u)udu = 0, \\ hy^{(1)}(x)\int w(u)udu, \ \text{if } \int w(u)udu \neq 0. \end{cases} \tag{4.52}$$

Comments to Proposition 4.

1. We use two formulas for the main term of the bias $m_e(x,h)$ as $h \to 0$. If $w(u) = w(-u)$ we have $\int w(u)udu = 0$, then the bias has the order h^2 and it is proportional to the second derivative $y^{(2)}(x)$ of the signal. If the $w(u)$ is not even then the bias has the order h and it is proportional to the first derivative $y^{(1)}(x)$.

2. For the median estimator $F(x) = |x|$, $F^{(1)}(x) = sign(x)$ and $F^{(2)}(x) = 2\delta(x)$ and we obtain from (4.42) that

$$V = \frac{1}{4(p_\varepsilon(0))^2}, \tag{4.53}$$

In particular, for Gaussian random errors

$$V = \frac{\pi}{2}\sigma^2. \tag{4.54}$$

Substituting (4.54) in Proposition 4 proves the formula (4.10) used in the *ICI* rule for the estimate standard deviation.

3. Provided some nonrestrictive additional assumptions the asymptotic normality can be proved for the considered median estimates. These results can be revealed from the technique developed in [119] as well as mathematical details concerning the limit passages in the formulas (4.46) and (4.48).

The optimization of the window function produced in [119] gives the triangular function

$$w(x) = 1 - |x|, \ |x| \leq 1,$$

as the optimal one over the class of the symmetric windows.

4. Proposition 4 is valid for any residual function F, provided $F(x) = F(-x)$ and quite nonrestrictive assumption for the probability density p_ε, which can be different from Gaussian. The function F and probability density p_ε influence the value of the parameter V only. In particular, it means that Proposition 4 is valid for a wide class of robust $M-$ estimates developed in [50].

5. Now let us consider the point-wise MSE risk $r(x,h)$ and the problem of the ideal window size selection. As it is follows from Proposition 4

$$r(x,h) \triangleq E(e(x,h))^2 = (a_d h^d y^{(d)}(x))^2 + V\Delta b/h, \tag{4.55}$$
$$b = \int w^2(u)du, \ a_d = \frac{1}{d!}\int w(u)u^d du,$$

where $d = 2$ for the even window w and $d = 1$ otherwise.

The minimizing risk $r(x,h)$ gives for the ideal values of the window size $h^*(x)$ and the ideal risk $r^*(x) = r(x, h^*(x))$:

$$h^*(x) = \left(\Delta \frac{Vb}{a_d^2(y^{(d)}(x))^2}\frac{1}{2d}\right)^{1/(2d+1)}, \tag{4.56}$$

$$r^*(x) = (\Delta \cdot V)^{2d/2d+1} \cdot (y^{(d)}(x))^{2/(2d+1)} A, \tag{4.57}$$
$$A = (1+2d)\,(a_d)^{2/(2d+1)}\,(\frac{b}{2d})^{2d/(2d+1)}.$$

These formulas show how the optimal accuracy depends on the parameters of the problem. In particular, for $\Delta = 1/n$, where n is a number of observation on the interval $[0,1]$ the formula (4.57) shows the ideal MSE convergence rate as $n \to \infty$

$$r^*(x) = \mathit{0}(n^{-2d/2d+1}). \tag{4.58}$$

It can be seen that for $h = h^*(x)$ a ratio of the bias to the standard deviation is a constant, which does not depends on the derivative of $y(x)$ as well as on x and it is given by the simple formula

$$m_e(x, h^*(x))/\sigma(x, h^*(x)) \triangleq \gamma = \sqrt{\frac{1}{2d}}. \tag{4.59}$$

Thus, (4.56) demonstrates that the ideal window size $h^*(x)$ depends on the smoothness of the signal as it is determined by its derivative $y^{(d)}(x)$ and the window size should be varying if the derivatives of the signal are varying essentially. (4.56) can be used for the plug-in selection of the window size as it was mentioned in the introduction.

It follows from the formulas for the bias and the variance that

$$m_e(x,h) \leq \sigma(x,h) \cdot \gamma, \text{ for } h \leq h^*(x). \tag{4.60}$$

6. Let the loss function be quadratic $F(x) = x^2/2$ and the noise be Gaussian $N(0,\sigma)$. Then $F^{(1)} = x$, $E\{(F^{(1)})^2\} = E\{x^2\} = \sigma^2$ and $F^{(2)} = 1$, $E\{F^{(2)}\} = 1$. It follows that $V(F, P_\varepsilon) = \sigma^2$. If we substitute $V(F, P_\varepsilon) = \sigma^2$ in Proposition 4, we obtain the accuracy results concerning the zero-power LPA estimation of $1D$ signal.

Comparing with Proposition 2 we may recognize common features of the corresponding statements.

4.4.2 Accuracy of ICI adaptive algorithm

The accuracy analysis produced in [37], [38] for the linear LPA estimates can be generalized to the adaptive nonlinear median estimates. In particular, combining of Proposition 4 with presented in [37], [38] convergence rate results gives

Proposition 5 *Let* $n = 1/\Delta \to \infty$, $\chi_{1-\alpha/2} = \beta\sqrt{\ln n}$, *with a large constant* $\beta > 0$, $(h_{i+1} - h_i) = \mathit{0}(\Delta)$,. *Then the adaptive estimate* $\hat{y}_{h^+(x_s)}(x_s)$ *has the* *following convergence rate*

$$E(y(x) - \hat{y}^{+}(x))^2 = \mathit{0}\left(\frac{\ln n}{n}\right)^{\frac{2d}{2d+1}}. \tag{4.61}$$

Thus, the order of the convergence rate $(\ln n/n)^{2d/(2d+1)}$ of the adaptive estimate using the ICI rule is different only in the factor $\ln n$ from the ideal risk (4.58), which is derived provided that the smoothness of the function (derivative $y^{(d)}(x)$) is known. This is a standard price for the adaptive estimation with unknown smoothness of the signal. It is emphasized that for estimating a function of unknown smoothness this factor cannot be eliminated. It means that the convergence rate given in (4.61) is optimal one and can not be improved.

4.4.3 Simulation for $1D$ signals

The performance of the developed median algorithms with the ICI adaptive windows is demonstrated on the set of different experiments. We show a performance of the adaptive median filters comparing the efficiency of the ICI rule for the left, right, symmetric varying windows and for the introduced combined estimators. As the reference algorithms we consider the symmetric median estimate with the ideal invariant window size obtained provided that the true signal is known. We compare the estimation accuracy also versus some adaptive wavelet estimators. These wavelet algorithms are applied as they are presented in WaveLab 802 (http://www-stat.stanford.edu/~waveLab).

We wish to show the advantage of the varying adaptive window length as well as of the combining of nonsymmetric and symmetric window estimates.

All signals are of the length $n = 1024$ and given on the segment $[0, 1]$, $\Delta = 1/n$. The test functions "Blocks" and "HeaviSine" are used in simulation experiments with signal-to-noise ratio equal to 7. These functions, noise and conditions of the Monte Carlo statistical modelling are exactly as given in [15].

The standard deviation of the median estimate for the ICI is calculated according to the formula (4.10)

$$\sigma(x, h) = \sigma\sqrt{\frac{\Delta\pi}{2h}\int w^2(u)du}, \tag{4.62}$$

where σ is estimated as

$$\hat{\sigma} = \{\text{median}(|z(x_s) - z_{s-1}| : s = 2, .., n)\}/(\sqrt{2}\cdot 0.6745). \tag{4.63}$$

It is assumed that

$$H = \{h_s | h_s = \Delta \cdot \lfloor 1.45^s \rfloor,\ s = 1, ..., 17\}. \tag{4.64}$$

Let us start from the analysis of the $RMSE$ and the generalized CV criteria I_{GCV} as functions of Γ. Figure 4.5 and Figure 4.6 present these results for all our algorithms and both considered test functions. First, note that all curves have a clear minimum on Γ. The minimum usually locates in the segment $[1.0 \div 2.0]$. This interval can be used as a rule of thumb for selection of Γ for the ICI rule.

Another important point, which we wish to emphasize is that the curves I_{GCV} as functions of Γ imitate a basic behavior of the $RMSE$ for the left, right and symmetric estimates. In particular, the minimums of the both curves are achieved for more less the same values of Γ. It confirms the basic use of the CV tests. The criteria function I_{GCV} can be applied for selection of Γ minimizing $RMSE$ on the base of observations only. However, the proposed I_{GCV} is much worse for the combined and weighted median estimates. We can see from Figures 4.5 and 4.6, that the curves I_{GCV} are not able to locate correctly minimums of $RMSE$ for the combined left-right, left-right-symmetric and left-right-weighted median estimates.

The accuracy estimation of the test functions is evaluated on the base of the Monte Carlo statistical experiments.

The $RMSE$ is calculated as

$$RMSE = \sqrt{\frac{1}{M}\sum_{j=1}^{M}\frac{1}{n}\sum_{s=1}^{n}(y(x_s) - \hat{y}^{+}_{(j)}(x_s))^2}, \tag{4.65}$$

where the average over $M = 50$ Monte Carlo simulation runs is calculated. The subscript (j) indicates the randomness of the estimate for every $j - th$ run.

The values of $RMSE$ are given in Tables 1 for the Gaussian noise. The following notation is used for the algorithms: "L", "R", " S" denote the basic left, right and symmetric adaptive window size algorithms, respectively;

"L & R", "L & R & S", and "L & R & W" denote the combined estimates as they are defined in (4.22), (4.21) and (4.23)-(4.24).

"InvarIdeal" corresponds to the results achieved by the symmetric median with the ideal invariant window. In order to find the ideal window size we calculate $RMSE$ as a function of h and the value given in Table 4.1 is the minimum value of $RMSE$. Thus, every window size is constant for all 50 runs of Monte Carlo simulations. "BestInvarIdeal" gives the value of $RMSE$ provided that the optimization on the invariant h is produced for every run of simulation. Thus, optimization on h in "InvarIdeal" produced after the averaging $RMSE$ on M Monte Carlo simulation runs, while in "BestInvarIdeal" it is done before this averaging. Recall that this optimization assumes that the true signal is known. Thus, "BestInvarIdeal" is the best possible result provided that h is constant for all x.

"WavPO" and "WavTI" indicate the $RMSE$ values obtained by the wavelet periodized orthogonal and shift invariant, respectively.

Let us analyze the figures in Table 4.1 concerning the test-signal "Blocks". While the accuracy of the separate left, right and symmetric adaptive medians are not good enough, the combining improves the accuracy in quite a valuable way. Both L&R&S and L&R&W estimate ($RMSE = 0.3512$ and 0.3860) give better results, than the algorithms "InvarIdeal" and "BestInvarIdeal" ($RMSE = 0.4978$ and 0.4853). The best accuracy ($RMSE = 0.2160$) is obtained by the algorithm L&R&W, where the Kaiser weight, $w = kaiser(x, \beta)$, with $\beta = 7$ is used.

In comparison with the wavelet estimates the algorithm L&R&W outperforms WavPO and give about the same results as WavTI. Curves presented in Figure 4.7 ($\Gamma = 2.0$ and $\beta = 7$) illustrate estimates and give general ideas of the comparative quality. It can be conclude that L & R & W gives a high accuracy estimate, which accurately delineates all jumps of the signal.

The signal "HeaviSine" is more difficult for the median algorithms because any median estimate assumes, by default, that the signal is nearly constant in the window. As a result for smooth varying signal small windows have to be applied. However, a valuable improvement in estimation was achieved by application of the smooth windows in the L&R&W estimate. The Kaiser window with $\beta = 30$ was used in this simulation

The corresponding $RMSE$ for L&R&W given in Table 4.1 is only slightly better that the accuracy of the median estimates with invariant ideal window sizes. In comparison with the wavelet estimates L&R&W outperforms the WavPO algorithm and gives $RMSE$ close to the one achieved by WavTI. Figure 4.8 ($\Gamma = 2.0$ and $\beta = 30$) presents illustrations of the different estimates. We wish to note that the L&R&W gives a good result and delineates all specific features of the HeavySine signal.

Table 4.1. RMSE for Gaussian noise observations

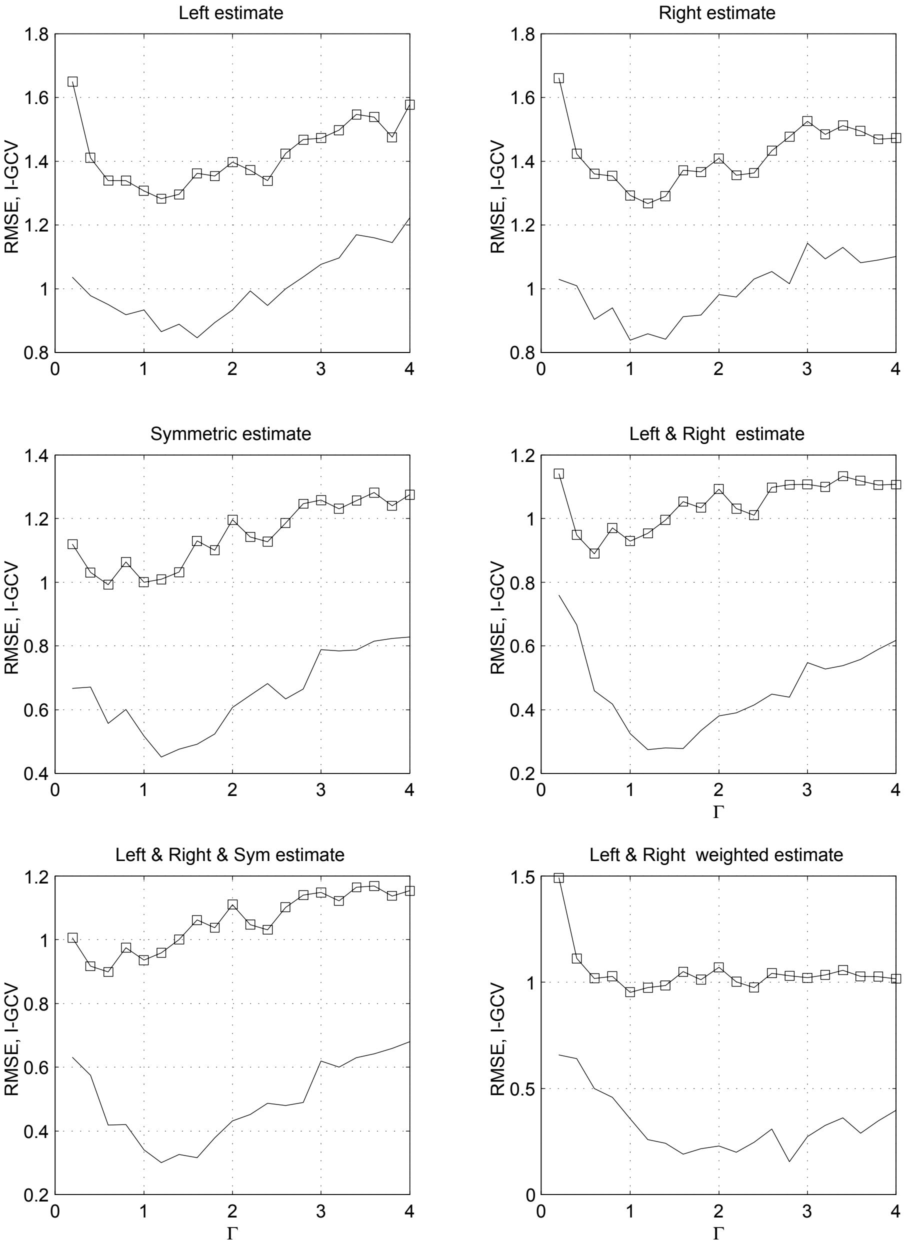
	Algorithm	"Blocks"	"HeavySine"
1	L	0.8879	1.0236
2	R	0.8785	1.0199
3	S	0.5470	0.4158
4	S&R	0.3512	0.5350
5	L&R&S	0.3860	0.4096
6	L&R&W	0.2160	0.3112
7	InvarIdeal	0.4978	0.3273
8	BestInvarIdeal	0.4853	0.3243
9	WavPO	0.3319	0.5568
10	WavTI	0.2140	0.2813

Curves in Figure 4.9 and 4.10 illustrate another important aspect of the median estimates, which are robust with respect to an impulse noise having a heavy-tailed distribution. It is assumed in simulation that the random noise has a mixed probability density

$$\begin{aligned} &f(x) = (1-\alpha)N(0,\sigma_1) + \alpha N(0,\sigma_2), \\ &\alpha = 0.05,\ \sigma_1 = 1,\ \sigma_2 = 100, \end{aligned} \tag{4.66}$$

with two Gaussian components having the standard deviations σ_1 and σ_2, respectively. Thus, $\alpha = 0.05$ determines a probability of high standard deviation components imitating rare and high amplitude impulses. Figures 4.9 and 4.10 show that the L&R&W filter is able to filter out this impulse noise quite efficiently. Here we wish to emphasize that in order to deal with the impulse noise a selection of the relevant grid H for the window size is important. Usually for the Gaussian noise we start in (4.64) from $s = 1$. Then, smallest numbers of observations are equal to 2 and 3 for the symmetric and nonsymmetric windows, respectively. In order to enable the algorithms to discard the impulse noise we start in (4.64) from $s = 3$. Then, respectively, the smallest numbers of observations are equal to 4 and 7 for the symmetric and nonsymmetric windows. For the "Blocks" larger values of the smallest window size increase a risk of miscalculation of the position of a jump into the signal and it can result in a shift of the estimates.

In Figure 4.10 the curves for HeaviSine are given. Note that the wavelet reference algorithms fail to deal with data contaminated by the heavy-tailed distribution noise.



Left estimate

Right estimate

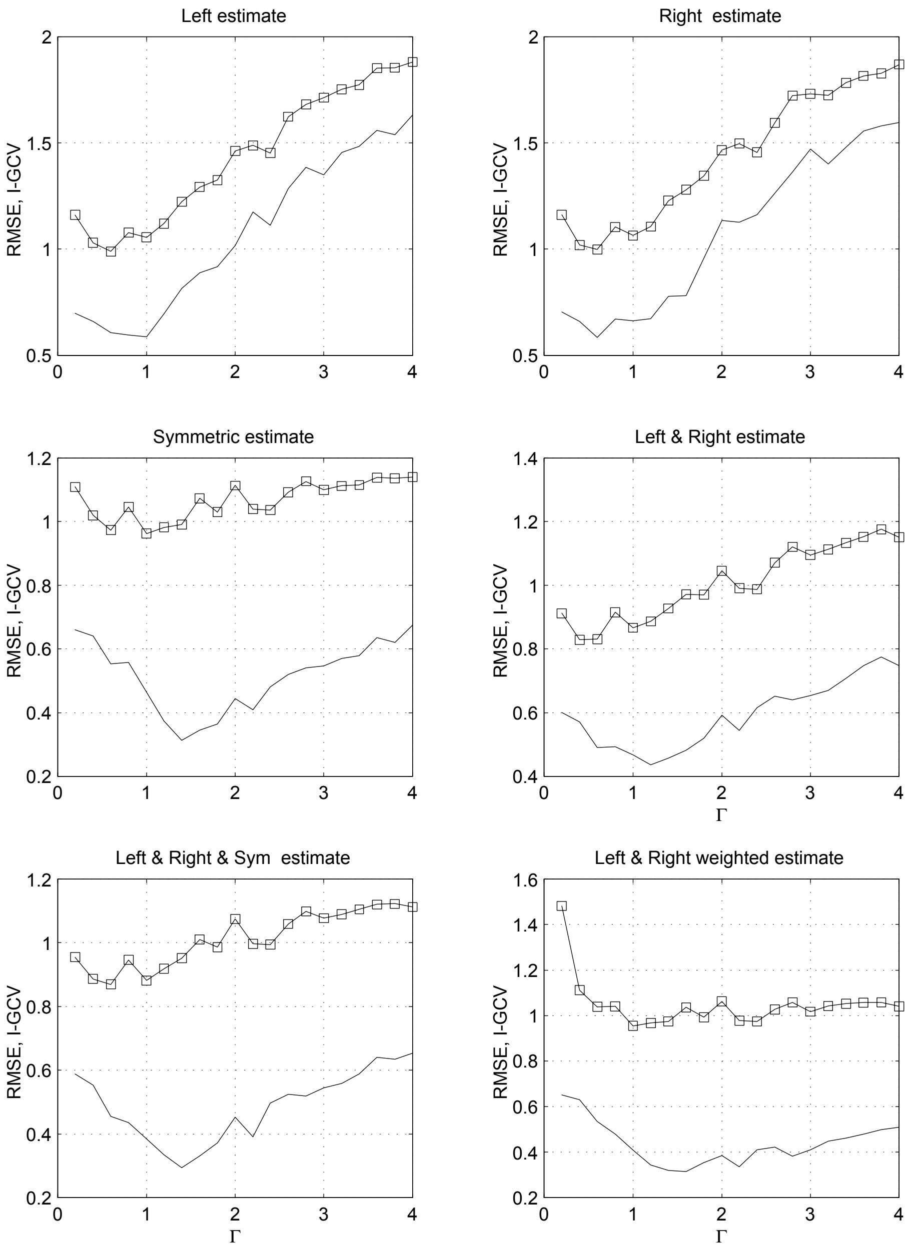
Symmetric estimate

Left & Right estimate

Left & Right & Sym estimate

Left & Right weighted estimate

Figure 4.5: Adaptive median estimates of "Blocks" signal. Curves present $RMSE$ (solid line) and I_{GCV} (cubes) versus the threshold parameter Γ for nonsymmetric, symmetric, combined and weighted median estimates.



Left estimate

Right estimate

Symmetric estimate

Left & Right estimate

Left & Right & Sym estimate

Left & Right weighted estimate

Figure 4.6: Adaptive median estimates of "HeaviSine" signal. Curves present $RMSE$ (solid line) and I_{GCV} (cubes) versus the threshold parameter Γ for nonsymmetric, symmetric, combined and weighted median estimates.

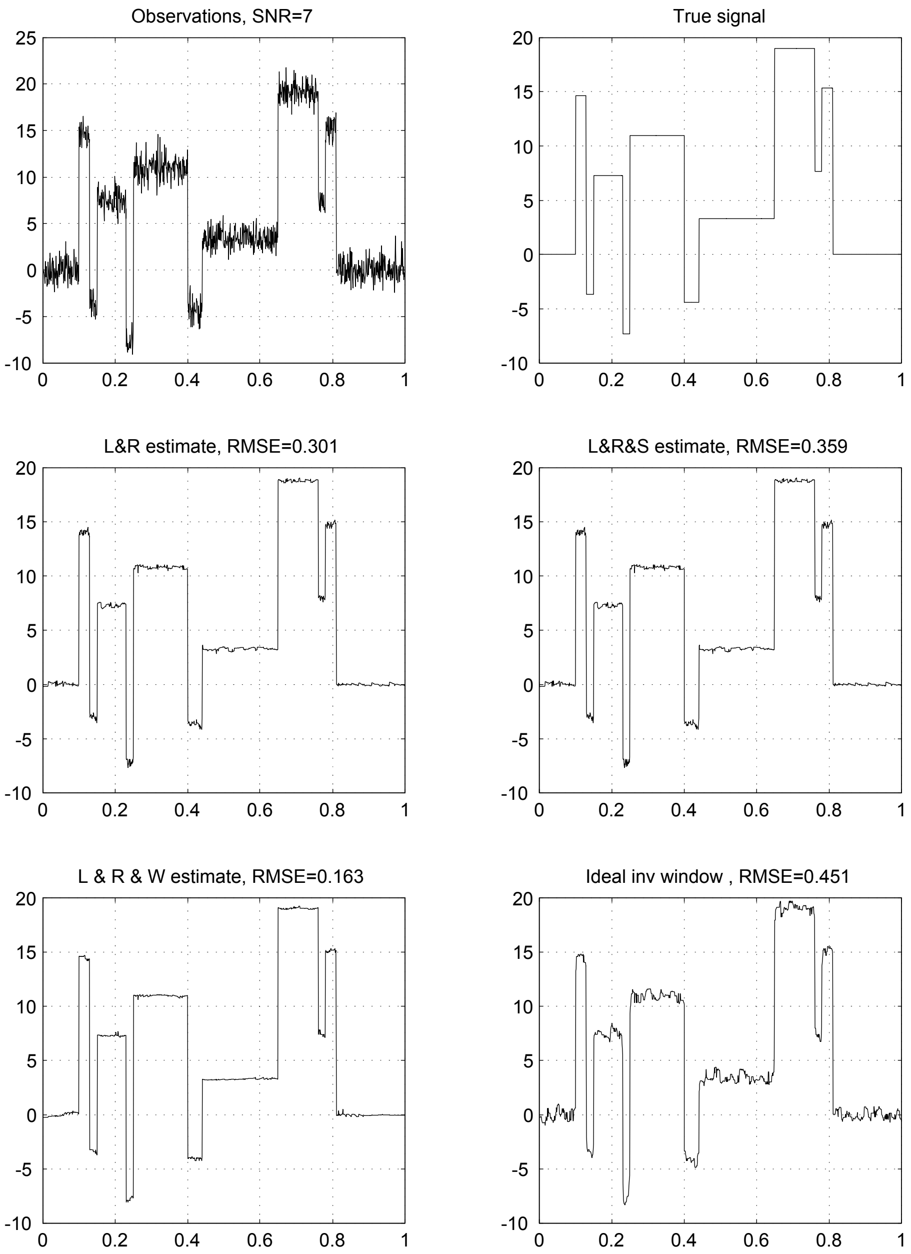


Figure 4.7: Adaptive estimates of ”Blocks” signal: a) Observations; b) True signal; c) Combined left and right estimates; d) Combined left, right and symmetric estimates; f) Weighted median estimate; e) Estimates with invariant ideal window size. For all estimates $\Gamma = 2$. Kaiser window with $\beta = 7$ is used.

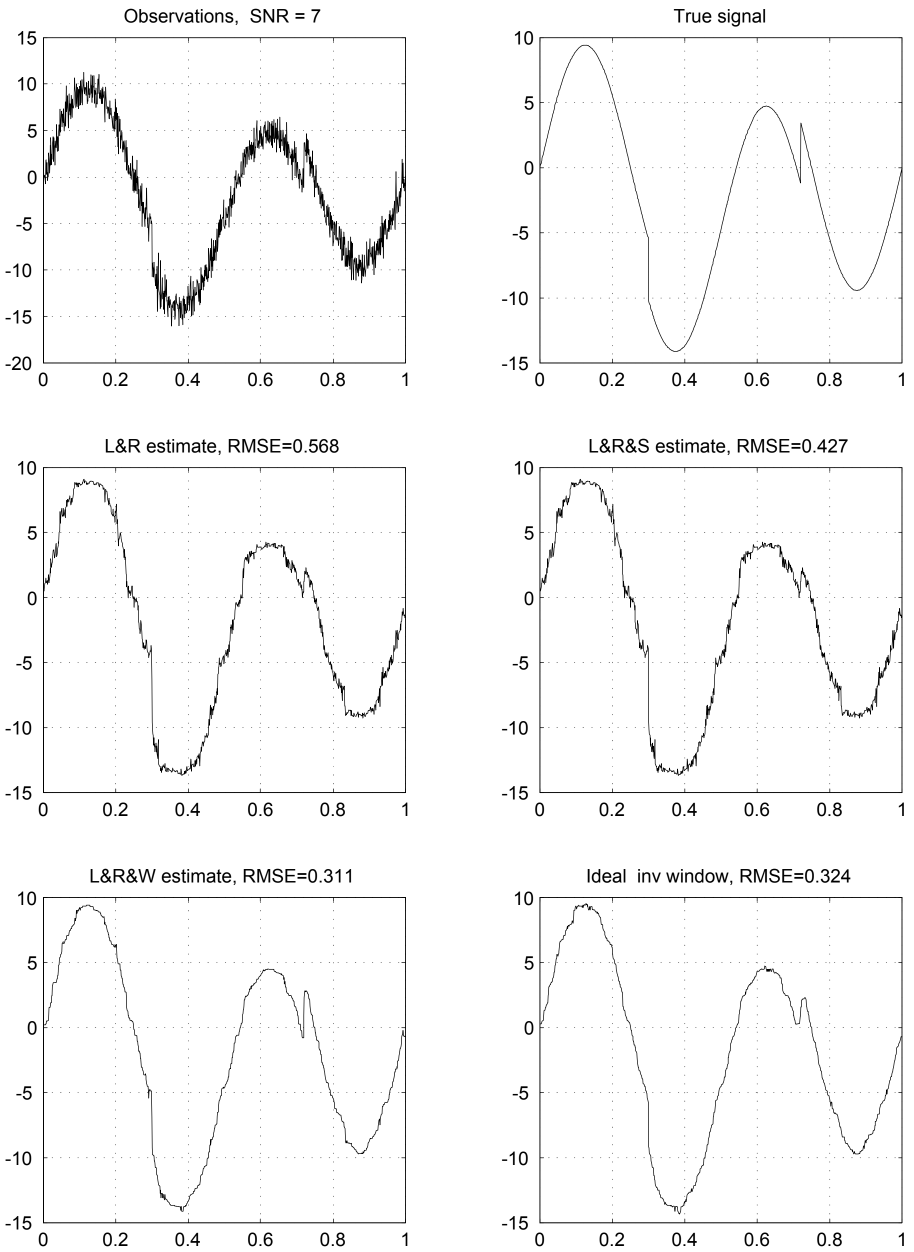


Figure 4.8: Adaptive estimates of "HeaviSine" signal: a) Observations; b) True signal; c) Combined left and right estimates; d) Combined left, right and symmetric estimates; f) Weighted median estimate; e) Estimates with invariant ideal window size. For all estimates $\Gamma = 2$. Kaiser window with $\beta = 30$ is used.

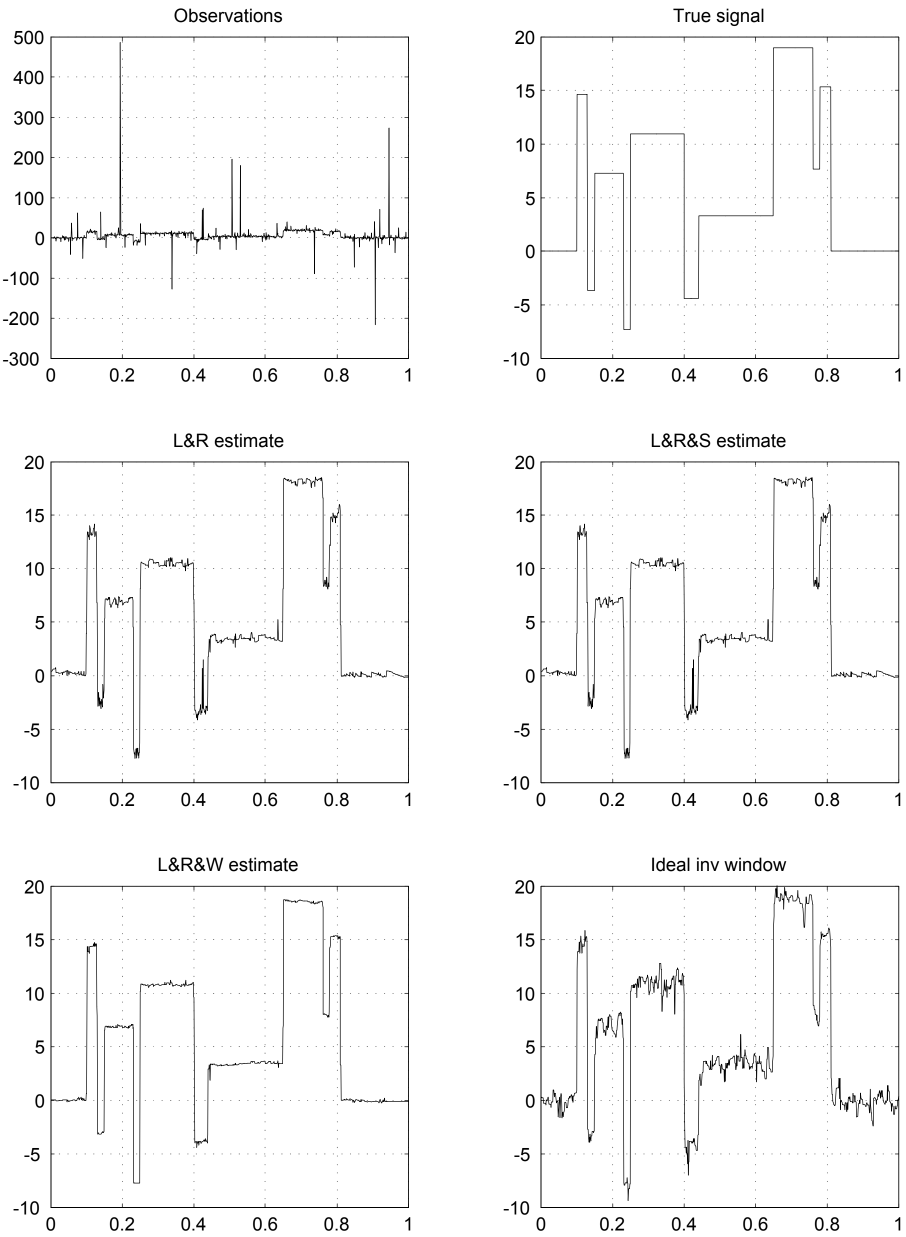


Figure 4.9: Adaptive estimates of "Blocks" signal contaminated by the heavy-tailed distribution noise: a) Observations; b) True signal; c) Combined left and right estimates; d) Combined left, right and symmetric estimates; f) Weighted median estimate; e) Estimates with invariant ideal window size. For all estimates $\Gamma = 2$. Kaiser window with $\beta = 7$ is used.

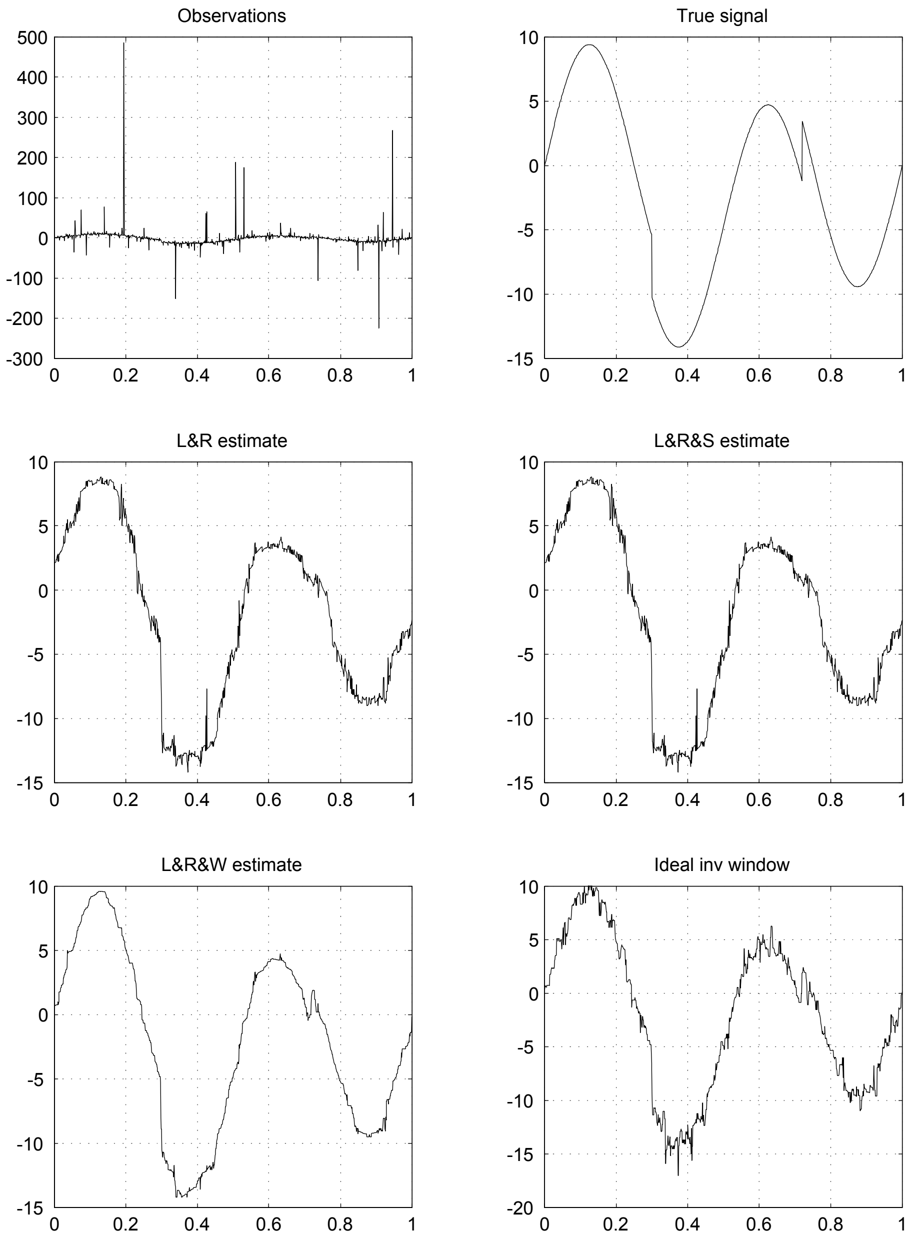


Figure 4.10: Adaptive estimates of "HeaviSine" signal contaminated by the heavy-tailed distribution noise: a) Observations; b) True signal; c) Combined left and right estimates; d) Combined left, right and symmetric estimates; f) Weighted median estimate; e) Estimates with invariant ideal window size. For all estimates $\Gamma = 2$. Kaiser window with $\beta = 30$ is used.

4.5 $2D$ image processing

The image processing based on the median estimates is completely similar to the $ICI - LPA$ adaptive algorithms considered in Chapter 1. Only two moments make a distinction between the ICI-median and $ICI - LPA$ methods.

The first moment concerns the median estimator itself. The LPA estimates naturally are replaced by the median, which can be the standard or weighted ones. The type of the used median is completely defined by the window w. If this w is uniform the median is standard calculated on the observations imbedded in this window. If the window w is not uniform the median is the weighted median calculated according to the algorithm (4.6)-(4.8).

The second moment concerns the ICI rule. The median estimate does not require serious changes in this window size selection algorithm presented in Chapter 1.

The only serious difference concerns the formulas used for calculation of the standard deviation of the estimates. For the $2D$ median estimates and the Gaussian noise this formula similar to given in Proposition 4 and in (4.62) has a form

$$\sigma(x,h) = \sigma \frac{\Delta}{h} \sqrt{\frac{\pi}{2} \int \int w^2(u_1,u_2) du_1 du_2}. \tag{4.67}$$

4.5.1 Simulation

Experimental results presented in this section are parallel to given in Chapter 3 for the LPA estimator with $m = 0$ and uniform window w. The scenarios of simulation experiments are completely identical with the only difference that here we use the median estimates instead of sample means exploited in Chapter 3. Experiments are performed for two test images: "Square" (8 bit binary 128×128 image) and "Cameraman" (8 bit gray-scale 256×256 image).

The results for the "Square" are shown in Figures 4.11 and 4.13. We can see the estimates are given by the symmetric window and quadrant median estimates as well as the final estimate obtained according to the equations (4.21)-(4.22). The ICI adaptive varying window sizes are shown in Figures 4.12 and 4.14. These results are given for two values of the threshold $\Gamma = 3.5$ and 2.0.

The results for the "Cameraman" are shown in Figures 4.15 and 4.17 for the symmetric window and quadrant median estimates as well as for the final estimate. The corresponding ICI adaptive varying window sizes are shown in Figures 4.16 and 4.18. These results are also given for $\Gamma = 3.5$ and 2.0.

The parallel $ICI - LPA$ results are shown in Figures 3.2, 3.3, 3.6, 3.7 for "Square" and in Figures 3.12, 3.13, 3.16, 3.31 for "Cameraman".

The comparison of curves and the criteria values demonstrates a remarkable similarity of the ICI-median and $ICI - LPA$ estimates. The varying adaptive window sizes have very similar shapes. The values of $RMSE$ and $ISNR$ have close values mainly in favor of the $ICI-LPA$ estimates at least for the final estimates.

It is interesting to note that some nonsymmetric median estimates give better results than the median estimates with the symmetric windows and the final estimates is always better than the quadrant and symmetric window estimates. Thus, fusing of the adaptive window estimates results in improvement of the image denoising.

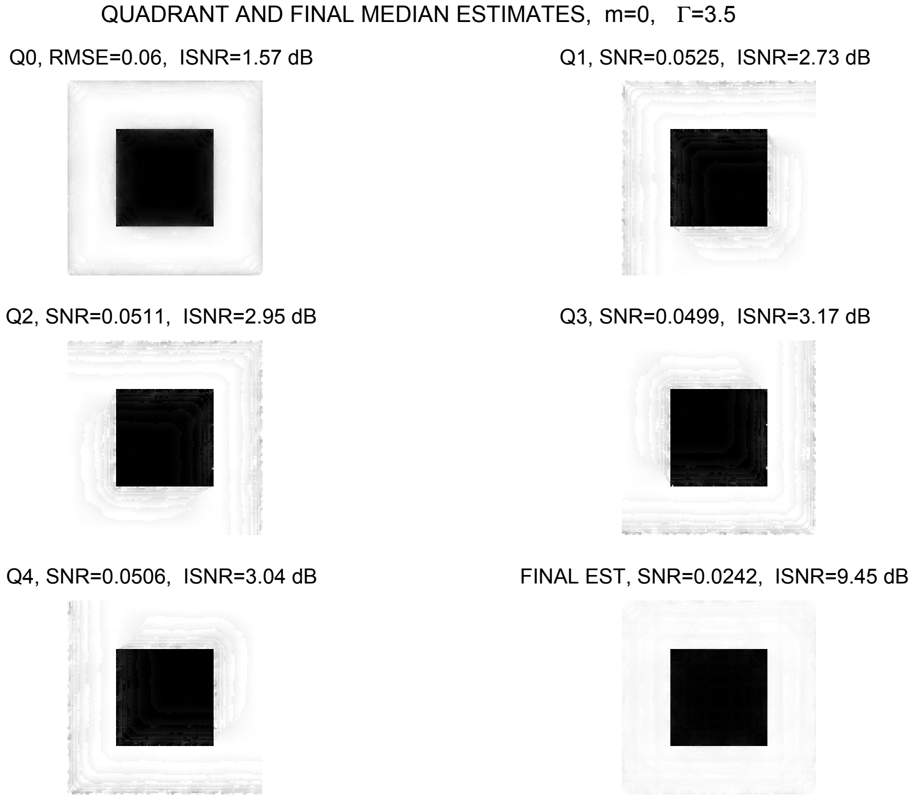


Figure 4.11: The ICI-median varying adaptive window size estimates. The estimates are quadrant, symmetric window and combined final, $\Gamma = 3.5$.

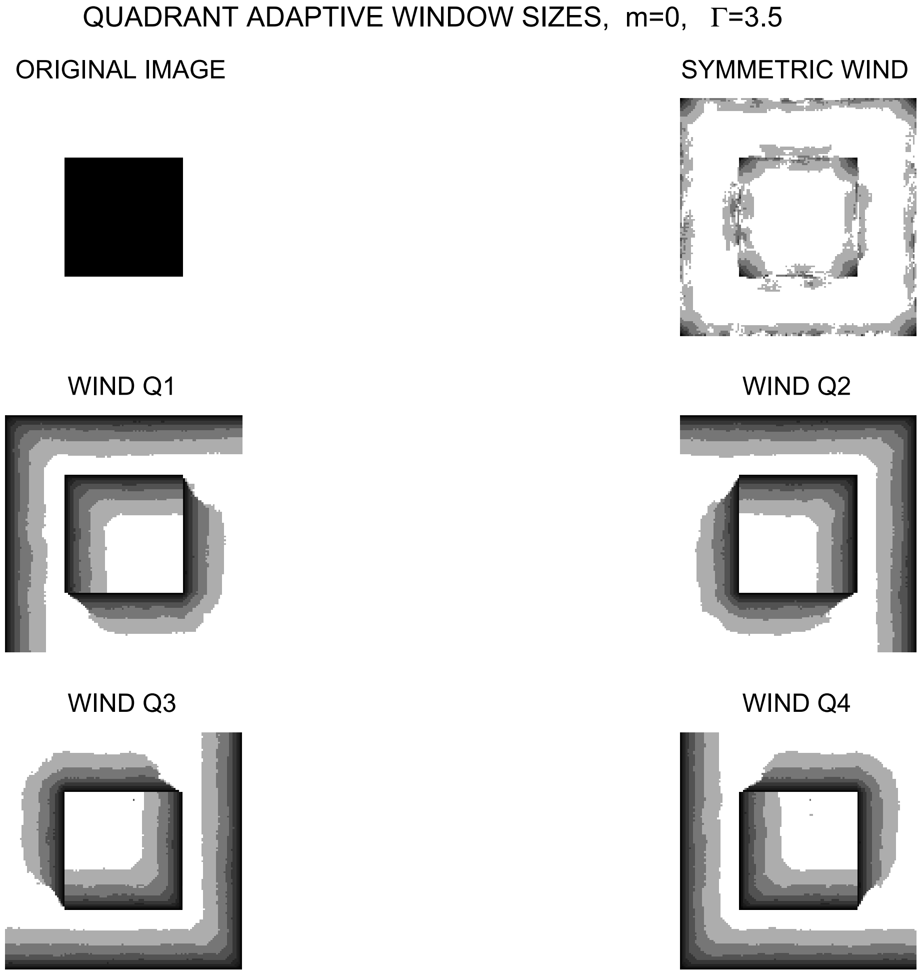


Figure 4.12: The *ICI*-median varying adaptive window sizes. The windows are quadrant and symmetric, $\Gamma = 3.5$.

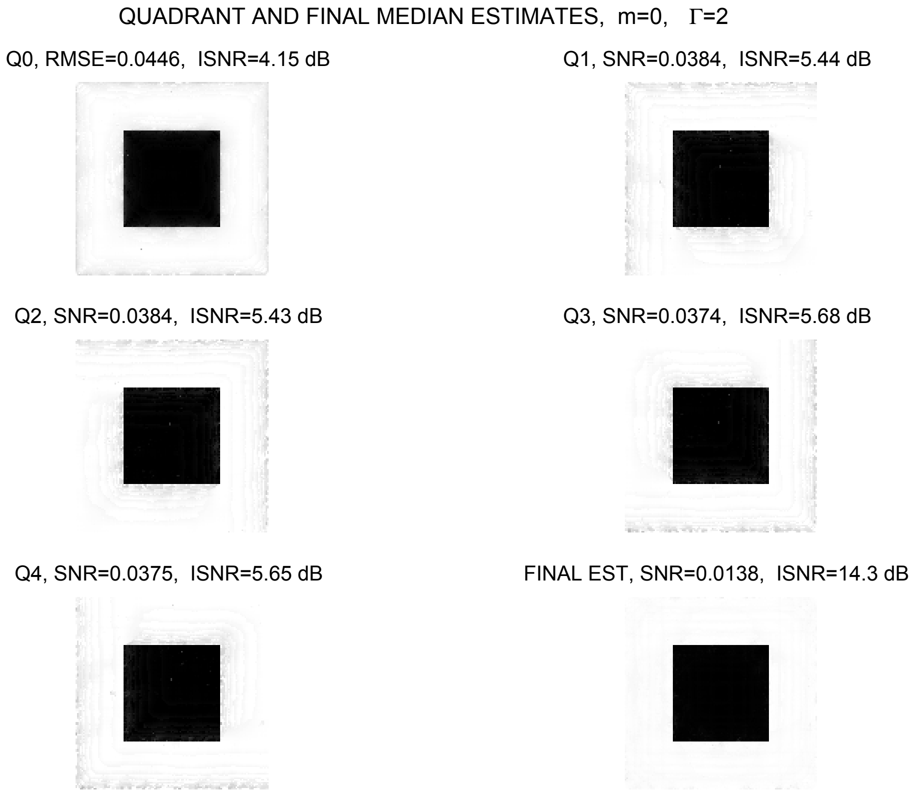


Figure 4.13: The ICI-median varying adaptive window size estimates. The estimates are quadrant, symmetric window and combined final, $\Gamma = 2.0$.

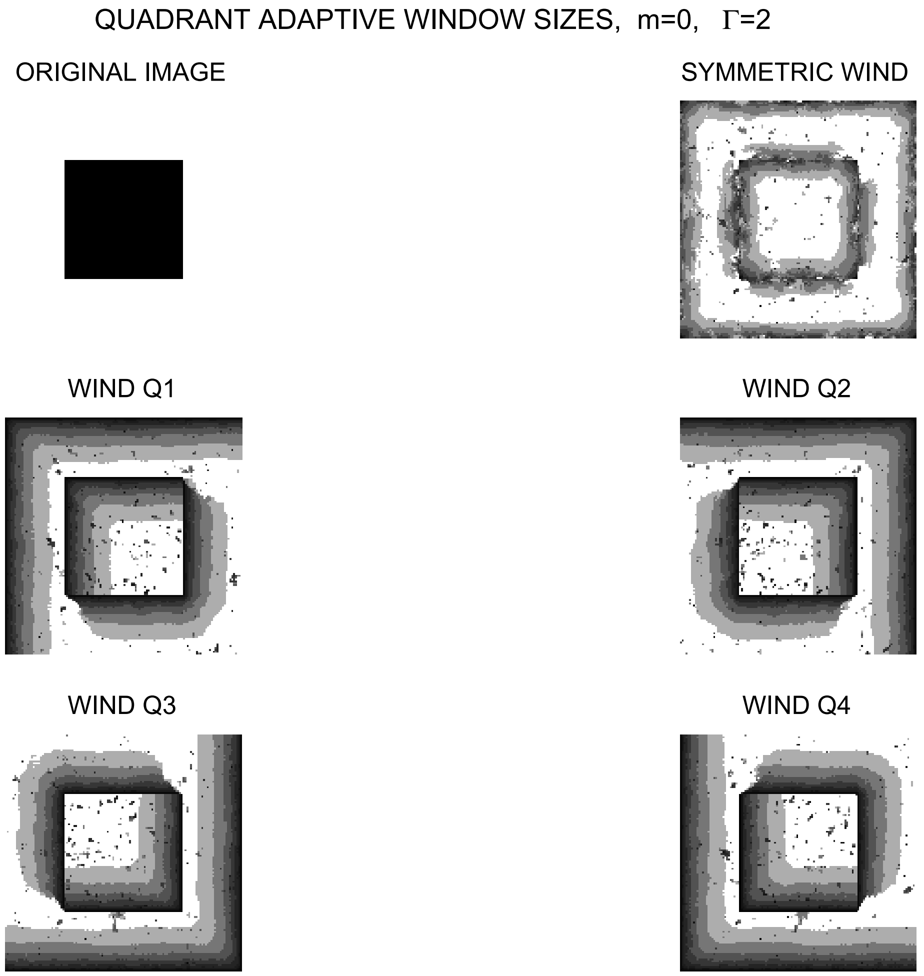


Figure 4.14: The ICI-median varying adaptive window sizes. The windows are quadrant and symmetric, $\Gamma = 2.0$.

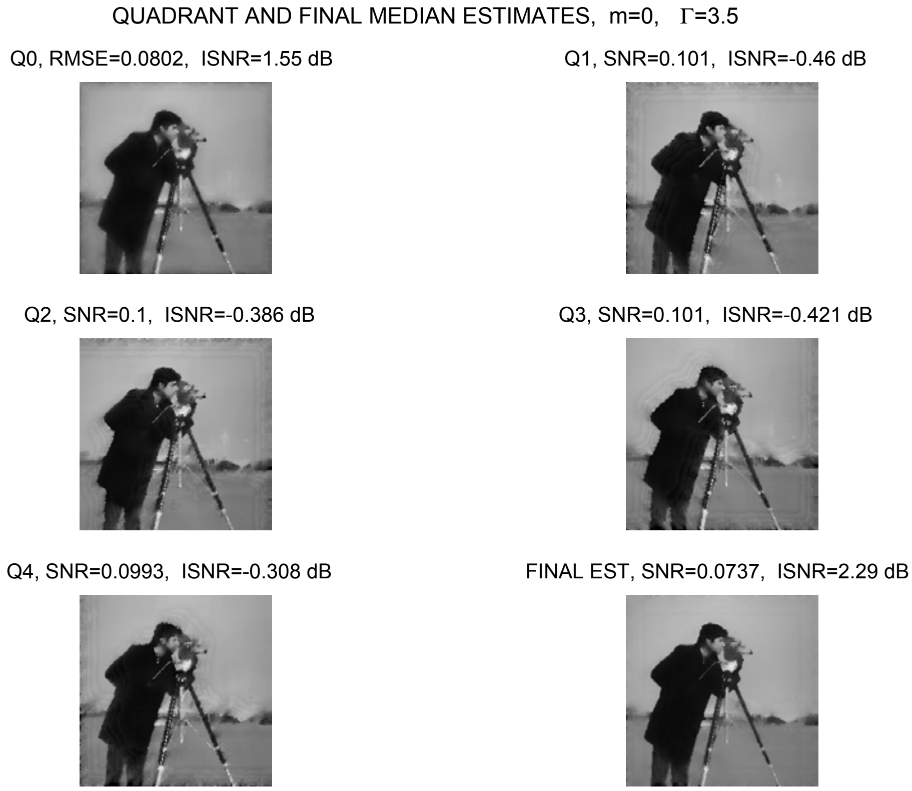

QUADRANT AND FINAL MEDIAN ESTIMATES, m=0, Γ=3.5
Q0, RMSE=0.0802, ISNR=1.55 dB
Q1, SNR=0.101, ISNR=-0.46 dB
Q2, SNR=0.1, ISNR=-0.386 dB
Q3, SNR=0.101, ISNR=-0.421 dB
Q4, SNR=0.0993, ISNR=-0.308 dB
FINAL EST, SNR=0.0737, ISNR=2.29 dB

Figure 4.15: The ICI-median varying adaptive window size estimates. The estimates are quadrant, symmetric window and combined final, $\Gamma = 3.5$.

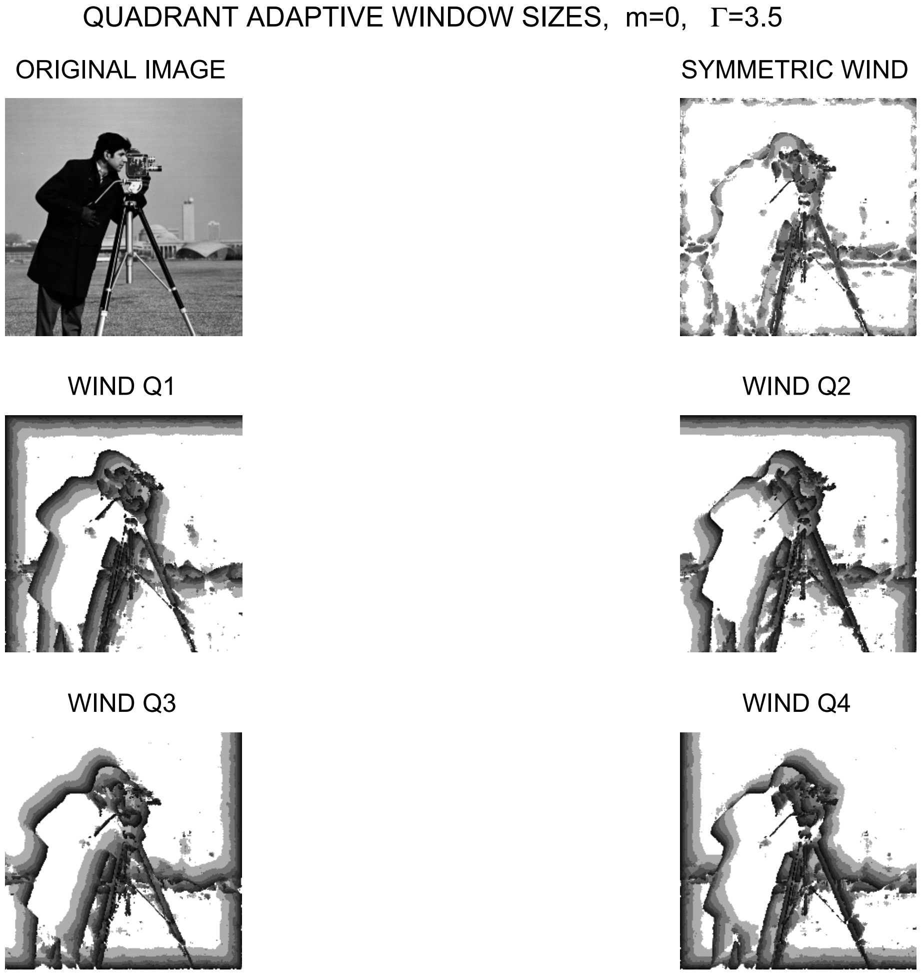


Figure 4.16: The ICI-median varying adaptive window sizes. The windows are quadrant and symmetric, $\Gamma = 3.5$.

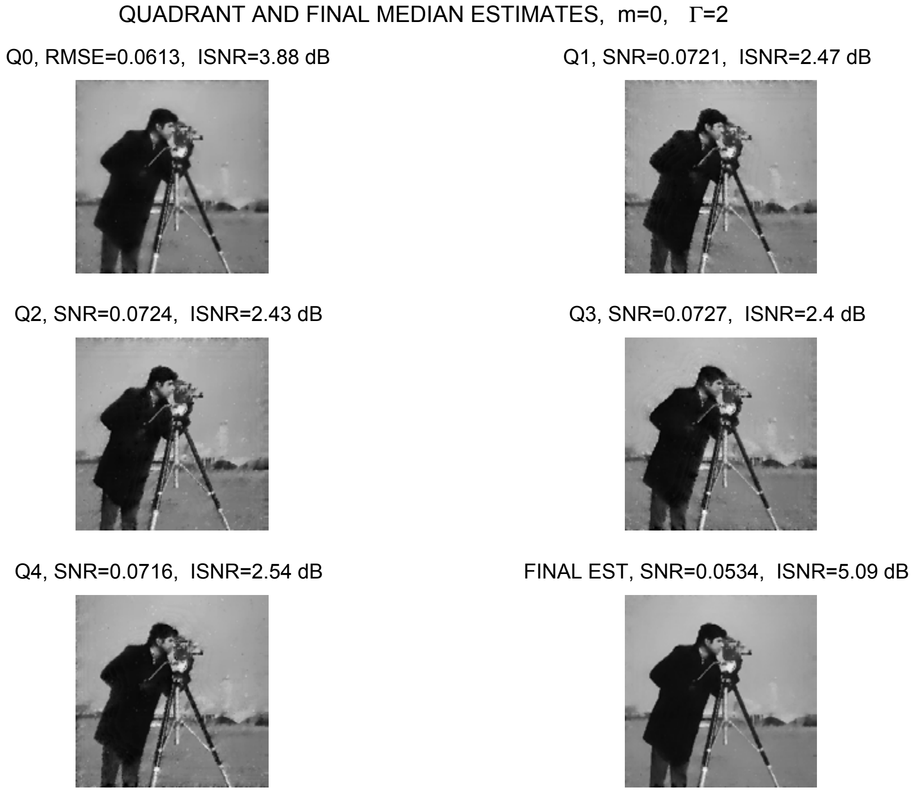

QUADRANT AND FINAL MEDIAN ESTIMATES, m=0, Γ=2
Q0, RMSE=0.0613, ISNR=3.88 dB
Q1, SNR=0.0721, ISNR=2.47 dB
Q2, SNR=0.0724, ISNR=2.43 dB
Q3, SNR=0.0727, ISNR=2.4 dB
Q4, SNR=0.0716, ISNR=2.54 dB
FINAL EST, SNR=0.0534, ISNR=5.09 dB

Figure 4.17: The ICI-median varying adaptive window size estimates. The estimates are quadrant, symmetric window and combined final, $\Gamma = 2.0$.

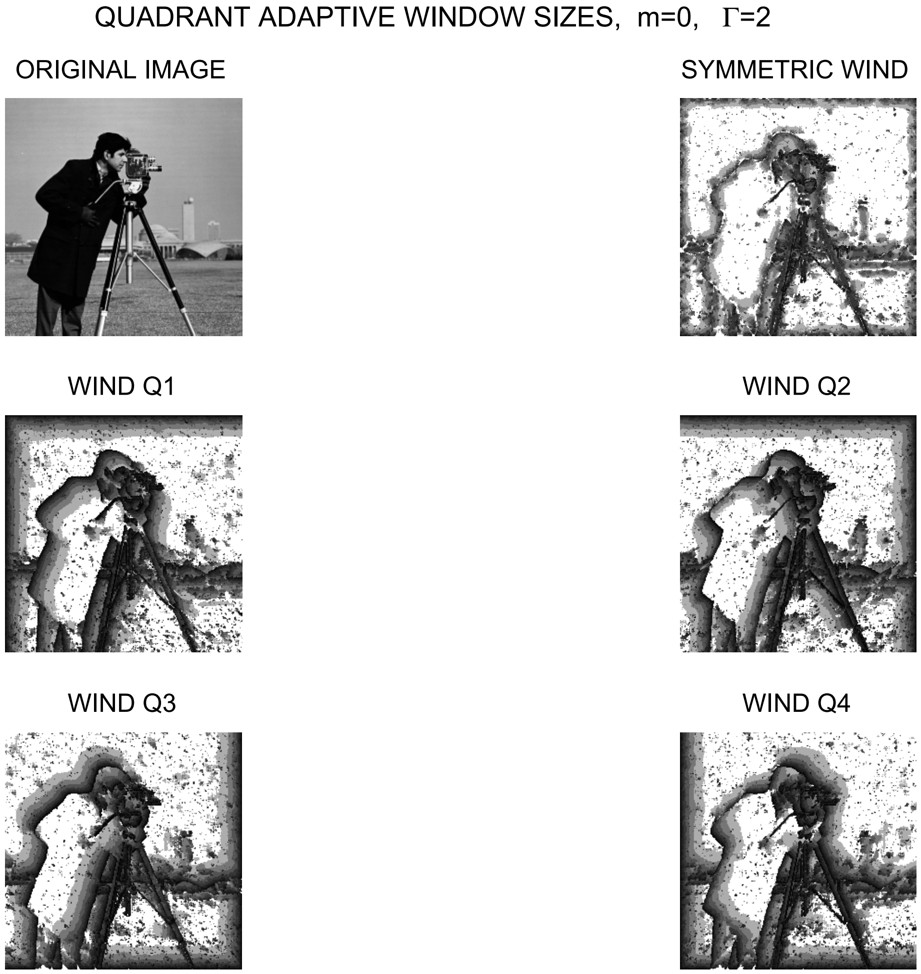


Figure 4.18: The ICI-median varying adaptive window sizes. The windows are quadrant and symmetric, $\Gamma = 2.0$.

4.6 LPA $2D$ discrete $M-$estimation

The accuracy analysis given in Proposition 4 can be extended to the $2D$ case in a straightforward manner. However, here we prefer to consider more general $2D$ estimators and to give more general results.

Let x be the center of the LPA and the estimate for the point x_s in the neighborhood of x is of the standard form (1.6)

$$\begin{aligned}
&y(x, x_s) = C^T \phi(x - x_s), \qquad (4.68)\\
&\phi(x) = (\phi_1(x), \phi_2(x), ..., \phi_M(x))^T,\\
&C = (C_1, C_2, ..., C_M)^T, \ x = (x_1, x_2), \ x_s = (x_{1,s}, x_{2,s}),
\end{aligned}$$

where $\phi(x) \in R^M$ is a vector of linear independent $2D$ polynomials of the powers from 0 till m.

However, what makes a difference with the LPA considered in Chapter 1, instead of (1.8) for estimation of C we use the nonquadratic loss function

$$J_h^F(x) = \sum_s w_h(x - x_s)F(z(x_s)) - y(x, x_s)), \tag{4.69}$$

where as usually the window $w_h(x) = w(x/h)/h^2$ formalizes the localization of fitting with respect to the centre x, while the scale parameter $h > 0$ determines the size of the window.

The loss function $F(x)$ in (4.69) is assumed to be symmetric, $F(x) = F(-x)$, as it is in Section 4.4.

Let the estimates of the vector C are found as a solution of the problem

$$\hat{C}(x, h) = \arg \min_{C \in R^M} \ J_h^F(x). \tag{4.70}$$

Then, it follows from (4.68) that

$$\hat{y}_h(x) = y(x, x_s)|_{x_s = x} = \hat{C}^T \phi(0) \tag{4.71}$$

and for the polynomials (1.7) it yields

$$\hat{y}_h(x) = y(x, x_s)|_{x_s = x} = C_1. \tag{4.72}$$

The LPA model (4.68) of the power m can be used for estimation of any derivative of the order k, $|k| \leq m$. According to the idea of the pointwise estimation discussed in Chapter 1, we derive these estimates in the form

$$\hat{y}_h^{(k)}(x) = \frac{\partial^{k_1 + k_2} y(x, x_s)|_{x_s = x}}{\partial x_{1,s}^{k_1} \partial x_{2,s}^{k_2}} = (-1)^{k_1 + k_2} \hat{C}^T \frac{\partial^{k_1 + k_2} \phi(0)}{\partial x_{1,s}^{k_1} \partial x_{2,s}^{k_2}}. \tag{4.73}$$

This definition of the derivative estimator assumes that differentiation in (4.73) is done with respect to x_s as $y(x, x_s)$ is an approximation considered as a function of x_s provided that the LPA center x is fixed and after that we assume that $x_s = x$.

For the polynomials (1.7) the derivative estimates (4.73) are simple:

$$\begin{aligned} &\hat{y}_h^{(1,0)}(x) = -\hat{C}_2, \hat{y}_h^{(0,1)}(x) = -\hat{C}_3, \hat{y}_h^{(2,0)}(x) = \hat{C}_4, \\ &\hat{y}_h^{(0,2)}(x) = \hat{C}_5, \hat{y}_h^{(1,1)}(x) = \hat{C}_6, \text{etc.} \end{aligned} \tag{4.74}$$

Thus, the coefficients of the LPA model (4.68) gives the estimates of the function and of the corresponding derivatives.

In this way, we arrive to the LPA $M-$estimates of the function and of the derivatives. It gives $2D$ generalization of the results given above in Section 4.4 for the LPA of the power larger than $m = 0$.

In the same time the estimates (4.69)-(4.74) can be treated as nonlinear LPA $M-$estimation version of the linear LPA estimates studied in Chapter 1.

For $F(x) = |x|$ the estimates (4.69)-(4.74) define the so-called L_1 nonparametric regression estimates, which are different from the M-median estimates by the higher power LPA used for more accurate fitting of the function in question.

The accuracy analysis technique demonstrated in Section 4.4 can be easily extended to the class of these more general estimates [59]. All results given Chapter 1 have the corresponding analogues for the LPA $M-$estimates with a simple replacement of σ^2 by V defined in (4.42).

Let us prove it.

Assume for $y(x)$ that:

(HM1) The image intensity $y(x)$ be deterministic locally smooth belonging to a nonparametric class of continuous $r-$differentiable functions (1.4) with $r = m + 1$

$$F_{m+1}(\bar{L}_{m+1}) = \{ \max_{r_1+r_2=m+1} \left| D^{(r)} y(x) \right| = L_{m+1}(x) \leq \bar{L}_{m+1}, \ \forall \ r_1 + r_2 = r \}, \tag{4.75}$$

where $\bar{L}_r$ is a finite constant;

(HM2) The LPA M-estimates of the intensity and the derivatives are defined by the equations (4.68) and (4.70). The power of the LPA is equal to m.

Assume for the criteria $F(x)$ and the noise distribution that:

(HM3) The random $\varepsilon(x_s)$ are independent and identically distributed for all x_s with the symmetric distribution

$$P_\varepsilon(x) = 1 - P_\varepsilon(-x). \tag{4.76}$$

(HM4) The loss function $F(x)$ is convex, bounded, twice differentiable almost everywhere and symmetric

$$F(x) = F(-x); \tag{4.77}$$

(HM5) The following holds

$$E\{F^{(1)}\} \triangleq \int F^{(1)}(v) dP_\varepsilon(v) = 0, \tag{4.78}$$

$$0 < E\{(F^{(1)})^2\} \triangleq \int (F^{(1)}(v))^2 dP_\varepsilon(v) < \infty, \tag{4.79}$$

$$0 < E\{F^{(2)}\} \triangleq \int F^{(2)}(v) dP_\varepsilon(v) < \infty, \tag{4.80}$$

It is assumed in the considered asymptotic that:
(HM6)

$$\Delta \to 0, \; h \to 0, \; h/\Delta \to \infty. \tag{4.81}$$

Proposition 6 *Let the assumptions (HM1)-(HM6) hold. Then the accuracy of the M-estimates (4.70) is characterized by the following asymptotic expressions:*

a) for the bias

$$\begin{aligned} |m_e(x,h)| \leq & \quad \bar{L}_{m+1} \sum_{r_1+r_2=m+1} \frac{1}{r_1! r_2!} \sum_s |g_h(x_s)| |x_{1,s}|^{r_1} |x_{2,s}|^{r_2}, (4.82) \\ |m_e^{(k)}(x,h)| \leq & \quad \bar{L}_{m+1} \sum_{r_1+r_2=m+1} \frac{1}{r_1! r_2!} \sum_s |g_h^{(k)}(x_s)| |x_{1,s}|^{r_1} |x_{2,s}|^{r_2}, \end{aligned}$$

and

b) for the variance

$$\begin{aligned} \sigma_y^2(x,h) &= E\{(\hat{y}_h(x) - E\{\hat{y}_h(x)\})^2\} = V(F,P_\varepsilon) \sum_s (g_h(x_s))^2, \quad (4.83) \\ \sigma_{y^{(k)}}^2(x,h) &= E\{(\hat{y}_h^{(k)}(x) - E\{\hat{y}_h^{(k)}(x)\})^2\} = V(F,P_\varepsilon) \sum_s (g_h^{(k_1,k_2)}(x_s))^2. \end{aligned}$$

Proof of Proposition 6.

Substitute (4.68) into (4.69) and represent the criteria function $J_h^F(x)$ in the form

$$\begin{aligned} & J_h^F(x) = \sum_s w_h(x-x_s) F(z(x_s)) - C^T \phi(x-x_s)) = \\ & \sum_s w_h(x-x_s) F(\varepsilon_s + y(x_s)) - C^T \phi(x-x_s)) = \\ & \sum_s w_h(x_s) F(\varepsilon_s + y(x-x_s)) - C^T \phi(x_s)). \end{aligned} \tag{4.84}$$

Using the Taylor series (1.155) for $y(x-x_s)$ gives

$$\begin{aligned} & y(x-x_s) = \sum_{k=0}^{m} \sum_{r_1+r_2=k} \frac{(-1)^{r_1+r_2}}{r_1! r_2!} x_{1,s}^{r_1} x_{2,s}^{r_2} D^{(r_1,r_2)} y(x) + \quad (4.85) \\ & \sum_{r_1+r_2=m+1} \frac{(-1)^{r_1+r_2}}{r_1! r_2!} x_{1,s}^{r_1} x_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda_s x_s), \\ & 0 \leq \lambda_s \leq 1. \end{aligned}$$

Let C^* be a vector of the function $y(x)$ and the derivatives $(-1)^{r_1+r_2}D^{(r_1,r_2)}y(x)$, with the corresponding signs, ordered accordingly to the polynomials

$$\frac{1}{r_1!r_2!}x_{1,s}^{r_1}x_{2,s}^{r_2}$$

in the vector $\phi(x)$ in (1.7).

Then, (4.85) can be represented as

$$y(x - x_s) = C^{*T}(x)\phi(x_s) + \Delta y(x, x_s), \tag{4.86}$$

where

$$\Delta y(x, x_s) = \sum_{r_1+r_2=m+1} \frac{(-1)^{r_1+r_2}}{r_1!r_2!} x_{1,s}^{r_1}x_{2,s}^{r_2} D^{(r_1,r_2)}y(x - \lambda_s x_s). \tag{4.87}$$

Inserting (4.86) in (4.84) gives

$$\begin{aligned} J_h^F(x) &= \sum_s w_h(x_s)F(\varepsilon_s + \Delta y(x, x_s) + \Delta C^T\phi(x_s)), \qquad (4.88) \\ \Delta C &= C^* - C, \end{aligned}$$

where ΔC is a vector of estimation errors.

For the accuracy analysis purposes the minimization $J_h^F(x)$ on C in (4.70) can be replaced by minimization of (4.88) on ΔC. Then, for the estimation errors we obtain the following equation

$$\frac{\partial J_h^F(x)}{\partial \Delta C} = \sum_s w_h(x_s)F^{(1)}(\varepsilon_s + \Delta y(x, x_s) + \Delta C^T\phi(x_s))\phi(x_s) = 0. \tag{4.89}$$

It is assumed in the accuracy analysis that the estimation and approximation errors are small, i.e. $\Delta C^T\phi(x_s)$ and $\Delta y(x, x_s)$ are small. As a next step, we can linearize the equation (4.89) with respect to the small sum $\Delta y(x, x_s) + \Delta C^T\phi(x_s)$

$$\begin{aligned} &\frac{\partial J_h^F(x)}{\partial \Delta C} \simeq \qquad (4.90) \\ &\sum_s w_h(x_s)[F^{(1)}(\varepsilon_s) + F^{(2)}(\varepsilon_s)(\Delta y(x, x_s) + \Delta C^T\phi(x_s)]\phi(x_s) = 0. \end{aligned}$$

It can be shown that the random

$$\sum_s w_h(u_s)F^{(2)}(\varepsilon_s)(\Delta y(x, x_s) + \Delta C^T\phi(x_s))\phi(x_s)$$

in (4.90) converges in probability to its expectation.

Asymptotically it gives

$$\sum_s w_h(x_s)[F^{(1)}(\varepsilon_s) + E\{F^{(2)}(\varepsilon_s)\}(\Delta y(x, x_s) + \Delta C^T \phi(x_s)]\phi(x_s) = 0.$$

In this way, we derive the error equation which is linear with respect to the estimation errors

$$\begin{aligned} E\{F^{(2)}(\varepsilon)\} \sum_s w_h(x_s)\phi(x_s)\phi^T(x_s)\Delta C = \\ -\sum_s w_h(x_s)\phi(x_s)[F^{(1)}(\varepsilon_s) + E\{F^{(2)}(\varepsilon)\}\Delta y(x, x_s)]. \end{aligned} \tag{4.91}$$

Using the standard notation (1.28)

$$\Phi_h = \sum_s w_h(x_s)\phi(x_s)\phi^T(x_s)$$

the solution of (4.91) is

$$\Delta C = -\Phi_h^{-1} \sum_s w_h(x_s)\phi(x_s)[\frac{F^{(1)}(\varepsilon_s)}{E\{F^{(2)}(\varepsilon)\}} + \Delta y(x, x_s)]. \tag{4.92}$$

Let us also use the notations for the estimation kernels introduced in (1.28)-(1.29)

$$\begin{aligned} &g_h(x_s) = w_h(x_s)\phi^T(0)\Phi_h^{-1}\phi(x_s), \\ &g_h^{(k_1,k_2)}(x_s) = w_h(x_s)[D^{(k_1,k_2)}\phi^T(0)]\Phi_h^{-1}\phi(x_s). \end{aligned}$$

According to the idea of the LPA the estimates are defined as (4.71) and (4.73). Then,

$$\begin{aligned} \phi^T(0)C^* &= y(x), \\ (-1)^{k_1+k_2} D^{(k_1,k_2)}\phi^T(0)C^* &= D^{(k_1,k_2)}y(x) \end{aligned}$$

and the estimation errors can be written in the forms

$$\begin{aligned} \phi^T(0)\Delta C &= y(x) - \hat{y}_h(x) = \\ (-1)^{k_1+k_2} D^{(k_1,k_2)}\phi^T(0)\Delta C &= D^{(k_1,k_2)}y(x) - \hat{y}_h^{(k)}(x). \end{aligned}$$

Then multiplying the both sides of (4.92) by $\phi^T(0)$ and $D^{(k_1,k_2)}\phi^T(0)$ we find

$$e_y(x,h) = y(x) - \hat{y}_h(x) = -\sum_s g_h(x_s)[\frac{F^{(1)}(\varepsilon_s)}{E\{F^{(2)}(\varepsilon)\}} + \Delta y(x,x_s)] \qquad (4.93)$$

and

$$e_y^{(k)}(x,h) = D^{(k_1,k_2)}y(x) - \hat{y}_h^{(k)}(x) = \qquad (4.94)$$
$$-\sum_s g_h^{(k_1,k_2)}(x_s)[\frac{F^{(1)}(\varepsilon_s)}{E\{F^{(2)}(\varepsilon)\}} + F^{(2)}(\varepsilon_s)\Delta y(x,x_s)].$$

Because, $E\{F^{(1)}(\varepsilon_s)\} = 0$ the bias of the estimates is as follows

$$m_y(x,h) = E\{y(x) - \hat{y}_h(x)\} = -\sum_s g_h(x_s)\Delta y(x,x_s), \qquad (4.95)$$
$$m_{y^{(k)}}(x,h) = E\{D^{(k_1,k_2)}y(x) - \hat{y}_h^{(k)}(x)\} = \qquad (4.96)$$
$$-\sum_s g_h^{(k_1,k_2)}(x_s)\Delta y(x,x_s).$$

It follows from (4.93) and (4.94) that the variance of the estimates is defined as

$$\sigma_y^2(x,h) = E\{(\hat{y}_h(x) - E\{\hat{y}_h(x)\})^2\} = \qquad (4.97)$$
$$V(F,P_\varepsilon)\sum_s (g_h(x_s))^2,$$
$$\sigma_{y^{(k)}}^2(x,h) = E\{(\hat{y}_h^{(k)}(x) - E\{\hat{y}_h^{(k)}(x)\})^2\} = \qquad (4.98)$$
$$V(F,P_\varepsilon)\sum_s (g_h^{(k_1,k_2)}(x_s))^2,$$

where

$$V(F,P_\varepsilon) = \frac{E\{(F^{(1)})^2\}}{(E\{F^{(2)}\})^2}. \qquad (4.99)$$

The formulas (4.97)-(4.98) proves (4.83).

Substituting (4.87) in (4.95) and in (4.96) we have

$$m_e(x,h) = -\sum_s g_h(x_s) \cdot \sum_{r_1+r_2=m+1} \frac{(-1)^{r_1+r_2}}{r_1!r_2!} x_{1,s}^{r_1} x_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda x_s),$$

and

$$m_e^{(k)}(x,h) = -\sum_s g_h^{(k)}(x_s) \cdot \sum_{r_1+r_2=m+1} \frac{(-1)^{r_1+r_2}}{r_1!r_2!} x_{1,s}^{r_1} x_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda_s x_s).$$

The corresponding upper bounds of the bias are as follows

$$|m_e(x,h)| = \quad (4.100)$$

$$|\sum_s g_h(u_s) \cdot \sum_{r_1+r_2=m+1} \frac{(-1)^{r_1+r_2}}{r_1!r_2!} x_{1,s}^{r_1} x_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda x_s)| \leq$$

$$\bar{L}_{m+1} \sum_{r_1+r_2=m+1} \frac{1}{r_1!r_2!} \sum_s |g_h(x_s)||x_{1,s}|^{r_1} |x_{2,s}|^{r_2}$$

and

$$|m_e^{(k)}(x,h)| = |y_h^{(k)}(x) - E\{\hat{y}_h^{(k)}(x)\}| = \quad (4.101)$$

$$|\sum_s g_h^{(k)}(x_s) \cdot \sum_{r_1+r_2=m+1} \frac{(-1)^{r_1+r_2}}{r_1!r_2!} x_{1,s}^{r_1} x_{2,s}^{r_2} D^{(r_1,r_2)} y(x - \lambda_s x_s)| \leq$$

$$\bar{L}_{m+1} \sum_{r_1+r_2=m+1} \frac{1}{r_1!r_2!} \sum_s |g_h^{(k)}(x_s)||x_{1,s}|^{r_1} |x_{2,s}|^{r_2}.$$

The formulas (4.100)-(4.101) proves (4.82). It completes the proof of Proposition 6.

Comparison of Proposition 6 with the analogous Proposition 2 for the linear LPA estimates we can conclude that all formulas and statements are identical provided that $V(F, P_\varepsilon)$ in Proposition 6 is replaced by σ^2 in Proposition 2. Actually it is a very interesting fact demonstrating similarity of the asymptotic properties of the linear and nonlinear M-estimates.

It is clear also that the asymptotic integral version of the accuracy analysis is valid for the estimates (4.70) in the form of Proposition 3 where σ^2 is replaced by $V(F, P_\varepsilon)$.

Chapter 5

DEBLURRING

5.1 Introduction

Suppose we wish to recover an image $y(x)$, $2D$ function, but we able to observe only about $(y \circledast v)(x)$, where v is the blurring kernel or point spread function (PSF) of a linear discrete circular convolution. The blurring phenomenon modelled by PSF v (continuous or discrete) is very evident in many image applications. Such linear inverse problems arise in optical systems, satellite imaging, radiometry, ultrasonic and magnetic resonance imaging, etc. Moreover, we assume that the data are noisy, so that we observe $z(x)$ given by

$$z(x) = (y \circledast v)(x) + \varepsilon(x), \tag{5.1}$$

where x defined on the regular $n_1 \times n_2$ lattice

$$X = \{(k_1, k_2),\ k_1 = 1, 2, ..., n_1,\ k_2 = 1, 2, ..., n_2\}$$

and ε is zero mean white Gaussian noise with the variance σ^2. We are interested in recovering the image $y(x)$ from the data $z(x)$ which are distorted by the convolution as well as by the additive noise. It is assumed that the PSF v is known.

Thus in this chapter we consider the problem with the observation model different from (1.1).

In the discrete Fourier transform (DFT) domain we equivalently have for (5.1)

$$Z(f) = Y(f)V(f) + \varepsilon(f), \tag{5.2}$$

with $Z(f), Y(f), V(f)$ and $\varepsilon(f)$ of the respective size $n_1 \times n_2$ DFT and the $2D$ normalized frequency $f \in F$,

$$\begin{aligned} F = \{&(f_1, f_2),\ f_1 = 2\pi k_1/n_1,\ f_2 = 2\pi k_2/n_2, \\ &k_1 = 0, 1, ..., n_1 - 1,\ k_2 = 0, 1, ..., n_2 - 1\}. \end{aligned} \tag{5.3}$$

An unbiased estimate of Y can be obtained through the straightforward solution of the equation (5.2) as the pure inverse of (5.2)

$$\hat{Y}(f) = \frac{1}{V(f)} Z(f). \tag{5.4}$$

However, in the cases of most interest, v is not invertible, in the sense that the inverse transform does not exist as a bounded linear operator; such inverse problems are called ill-posed or ill-conditioned (e.g. [6], [101], [117] and [122]). In the frequency domain the ill-posed means that $V(f)$ may take zero or close to zero values. If the system is ill-posed, then the random component in the estimate is enormously amplified in the inversion to yield an extremely noisy useless estimate.

It is now standard to approach inverse problems by the method of regularization, in which one applies, rather than the inversion, a regularized inverse operator. It gives instead of (5.4)

$$\hat{Y}(f) = \frac{V(-f)}{|V(f)|^2 + \varepsilon^2} Z(f), \tag{5.5}$$

where $\varepsilon > 0$ is a regularization parameter.

This typically produces a reconstruction in which certain features of the original image are 'smoothed away' and it seems to be a blurred version of the original. This blurring phenomenon is particularly of concern when the underlying object has edges, and when the location and size of these edges are of central interest. Such edge-dominated situations are relatively common in imaging applications, where the edges signify boundaries between different parts of a scene or different layers in the earth, or different organs in the body. It would be of interest to obtain sharper reconstructions for objects with edges.

The phenomenon of blurring and the goal of edge recovery have been studied by many researchers over the last few years; a partial listing of articles specifically devoted to this theme would include [10], [27], [28], [116].

One of the regular approaches to the problem is based on projection methods assuming that the image in question can be decomposed in sums of basic functions. A natural way of projection for ill-posed problems is associated with the singular value decomposition (SVD). Denote v^* the adjoint of v and assume that v^*v is an operator with the eigenvalues $\{b_k^2\}$, $b_k > 0$, and with eigenfunction $\{\varphi_k\}$. These eigenfunctions are used as basic functions in SVD methods (e.g. [6], [8], [9] and references herein).

Despite the great popularity of schemes based explicitly or implicitly on SVD, the method suffers from natural performance limitations. These are rooted in the fact that the basis functions $\{\varphi_k\}$ derive from the convolution operator v under study, not from the object y to be recovered. Thus, if the same operator v occurs in two different fields of scientific inquiry, the basis functions will be the same, even though the type of object to be recovered may be quite different in the two fields. Efficient representation of the object y by the eigenfunction functions of v^*v, however, cannot be guaranteed [8].

The SVD focuses exclusively on properties of the forward convolution operator rather than the object to be recovered. In essence, it diagonalizes this operator. There are important alternative strategies based on developing basic functions for the object y rather than the convolution operator v. Wavelets provide an effective solution to the problem. Many real-world images and signals have economical - domain representations in which a few large wavelet coefficients capture most of the signal energy. Signal and image reconstruction using wavelet shrinkage is a spatially adaptive process well suited to signals with edges and other singularities [84], [15], [1], [16], [54].

The wavelet - vaguelette decomposition has been proposed for the deconvolution problem in [17]. In this approach the wavelet series is applied to the signal obtained as an output of the pure deconvolution (5.4). Scale-dependent shrinkage is employed to estimate the signal wavelet coefficients. It was proved that the wavelet - vaguelette technique is nearly optimal to recover a wide class of signals.

Recently a new hybrid wavelet based deconvolution algorithm was proposed in [100]. This algorithm comprises Fourier-domain regularized inversion followed by wavelet-domain noise suppression. It is shown that the developed algorithm outperforms the conventional Wiener filter and wavelet-based image restoration algorithms.

Further development of the wavelet approach is proposed in [8], where a new decomposition which is much better adapted to the type of edge-dominated object while providing an almost diagonal representation of the Radon convolution operator. It is done by introducing new base functions, so-called curvelets: smooth, highly anisotropic directional elements well suited for detecting and synthesizing curved edges.

A special common point of all methods discussed above starting from the frequency domain equation (5.2) is that some basis functions are introduced and applied for approximation of the object function $y(x)$ in the form of series with coefficient defined from the observations. These functions may be Fourier harmonics as in (5.4) and (5.5), eigenfunctions of the convolution operator in SVD methods or wavelets in wavelet based decompositions. There exists a lot deconvolution techniques based on this sort of approaches.

Essentially different ideas and methods arise from a pointwise nonparametric estimation approach considered in this book. These methods mostly do not assume any underlying global parametric model of the object $y(x)$ and do not use the parametric series for object approximation. It is assumed only that the object is composed from piecewise smooth elements and every point of the object allows a good local approximation. Actually the methods are based on kernel smoothing with a special choice of the kernels. The main goal of estimation is to build a pointwise approximation using the observations from a neighborhood. The estimation is produced in a sliding mode what means that the every point of the object is treated in a special way in order to achieve the best possible performance. There is a number of proposals for nonparametric smoothing of images which allow for preserving the sharp edge structure as well as the edge detection and reconstruction in direct observations.

This sort of methods have been studied in the previous chapters of this book. In this chapter we consider the development of the nonparametric LPA approach for the image deblurring problems.

We show also that the difficulty of nonparametric image deblurring depends heavily on the smoothness of PSF: the smoother PSF is, the harder the deblurring will be.

The smoothness of PSF is defined by asymptotic behavior of the integral Fourier transform of $v(x)$

$$V(\lambda) = \int_{R^2} v(x) \exp(-i2\pi < \lambda, x >) dx,$$

as $||\lambda|| = \sqrt{\lambda_1^2 + \lambda_2^2} \to \infty$, where $\lambda = (\lambda_1, \lambda_2)$ and $< \lambda, x >= \lambda_1 x_1 + \lambda_2 x_2$.

We call the PSF ordinary smooth (or polynomial decaying) of the order α if its frequency characteristic $V(\lambda)$ satisfies

$$|V(\lambda)| \simeq c_0 ||\lambda||^{-\alpha}, \ \ ||\lambda|| \to \infty$$

and we will call the PSF super smooth (or exponential decaying) of the order α if its frequency characteristic $V(\lambda)$ satisfies

$$|V(\lambda)| \simeq c_0 \exp\{-\gamma ||\lambda||^{\alpha}\}, \ , \ ||\lambda|| \to \infty.$$

If $v \geq 0$ and $\int_{R^2} v(x)dx = 1$ then v can be treated as a probability density function.

The examples of ordinary smooth $PSFs$ include uniform, gamma, double exponential and symmetric probability density functions.

The examples of super smooth $PSFs$ are normal, mixture normal, Cauchy and Lévy "stable" with the density

$$V(\lambda) = \exp\{-\gamma ||\lambda||^{\alpha}\}, \gamma > 0, \ 0 < \alpha \leq 2.$$

Such Lévy PSF appears in many important image deblurring applications [13]. The Gaussian case, corresponding $\alpha = 2$, occurs in very diverse contexts including optical seekers in cruise missiles, undersea imaging, nuclear medicine, computed tomography, and ultrasonic imaging in nondestructive testing. The case $\alpha = 5/3$ describes atmospheric turbulence blurring and is important in some astronomical and surveillance satellite applications. The case $\alpha = 1$ corresponds to the Cauchy or Lorentzian distributions. This has been used to model X-ray scattering in radiology. Values of α satisfying $0 < \alpha \leq 2$ characterize a wide variety of electronic-optical devices. Such devises are important in night vision systems and many biomedical imaging systems. References for many of these application of the Lévy densities may be found in the bibliographies of [11], [12], [13]. In some applications, several electron-optical devices may be cascaded together and used to image objects through a distorting medium such as the atmosphere or the ocean, or may be combined to produce biomedical tool.

The overall PSF is then the convolution product of the individual components PSF with

$$V(\lambda) = \exp\{-\sum_{k} \gamma_k ||\lambda||^{\alpha_k}\},\ \gamma_k > 0,\ 0 < \alpha_k \leq 2. \tag{5.8}$$

The general functional form (5.8) may also be used to best-fit a large class of empirical determined optical transfer functions, by suitable choices of the parameters γ_k and α_k [13].

A nonparametric deconvolution of the super smooth v is a more difficult problem than the deconvolution of the ordinary smooth v. While the convergence rate for deblurring of the ordinary smooth PSF is proportional to $1/n^{\beta_0}$, where n is a number of data and β_0 is a positive constant, this rate for the super smooth is of the order $(1/\log n)^{\beta_1}$, $\beta_1 > 0$. Thus, in the convergence we have $\log n$ versus n. It means a principal lower convergence rate for the super smooth PSF.

The considered nonparametric deconvolution originated in probability density estimation provided that a desired signal is observed with an additive noise. In this problem the observation density is a convolution of the noise and the desired signal density. If the noise density is known the estimation of the desired signal density involves the deconvolution operation with a known PSF. The concepts of the ordinary and super smooth PSF (probability densities) as well as a difference in the convergence rate originally have been obtained for the problems involving probability density deblurring [21], [22]. In this chapter we adapt and extend these results for image deblurring.

As it was discussed in Chapters 1 and 2 the adaptation is now commonly considered as a crucial element of nonparametric estimation. The adaptation methods even for originally linear estimates are finalized in nonlinear estimators. In this chapter we apply for the deblurring the $LPA - ICI$ estimators studied in the previous chapters.

First application of this kind adaptive kernel estimator for indirect observations give by the convolution has been reported in [40] where the scale adaptive kernel estimate with the ICI rule is applied to the $1D$ signal obtained by the pure deblurring (5.4). The continues time model is used for the algorithm presentation as well as for the analysis. The accuracy performance produced shows that the estimator possesses the best possible ability for the pointwise adaptive estimation. In this chapter this approach is extended to $2D$ discrete data imaging as well as to data obtained as output of the regularized and Wiener inversion. The last case is quite similar to the hybrid filter studied in [100] with the essential difference that the adaptive kernel estimate is used instead of the wavelet filters.

5.2 Adaptive kernel estimate

Consider a linear discrete kernel estimator defined on the lattice X and given by the kernel $g_h(x)$, $x \in X \subset R^2$, with the scale parameter $h > 0$:

$$y_h(x) = (g_h \circledast y)(x) \tag{5.9}$$

Here $\circledast$ denotes the $2D$ convolution of the signal $y(x)$ and the kernel $g_h(x)$. Thus, $y_h(x)$ is an output of the filter defined by the weight function $g_h(x)$.

The following is assumed for $g_h(x)$:

(G1) The m order vanishing moments hold:

$$(g_h \circledast x^k)(0) = \delta_{|k|,0},\ |k| \leq m, \tag{5.10}$$

where $k = (k_1, k_2)$ is a multi-index, k_1, k_2 nonnegative integer, and $|k| = k_1 + k_2$, $x^k = x_1^{k_1} x_2^{k_2}$, $\delta_{|k|,0}$ is equal to one for $|k| = 0$ and to zero otherwise.

(G2)

$$||g_h||^2 = \sum_{x \in X} |g_h(x)|^2 \leq Bh^{-b},\ B,\ b > 0. \tag{5.11}$$

Then, we say that the g_h is a smoothing kernel estimator (low-pass filter) of the order m. The order means that the operator is reproductive with respect to $2D$ polynomial functions of the power less or equal to m, i.e. for a such polynomial $f(x)$

$$(g_h \circledast f)(x) = f(x). \tag{5.12}$$

The condition (5.11) means that the g_h is a low-pass filter with the bandpass narrowing as the scale h increasing. The norm $||g_h||^2$ defines the variance of the operator output with respect to the input white noise of the variance equal to 1.

The LPA as it is presented in Chapter 1 is a perfect tool for design of the kernels g_h satisfying the conditions (G1)-(G2). Remark that the window size h can also be interpreted as the scale parameter of the estimator.

If h is small the bandpass of the estimate is large and the signal $y(x)$ is reproduced by the filter without distortions. If h is large then the bandpass of the estimate is narrow and only low frequency components of $y(x)$ can be observed at output of the estimator. Thus for large h we obtain smooth signal $y_h(x)$ with suppressed high frequency features, while for small h all these small features and details are preserved . If the signal is given with an additive random noise then there exists an optimal value of the bandwidth corresponding to the usual trade-off between random and systematic errors of the signal reconstruction. The smoothing properties of the kernel $g_h(x)$ are well controlled by the parameter h. The super index h in notation $y_h(x)$ emphasizes that we deal with the smoothed version of the signal $y(x)$ and the level of smoothing involved is defined by value of h.

5.2.1 Basic idea

A basic idea of the proposed deblurring algorithm is to introduce the smoothed function $y_h(x) = (g_h \circledast y)(x)$ instead of the original $y(x)$ as a solution of the inverse problem and use the kernel estimator g_h equipped with the smoothing parameter h in order to suppress noise as much as possible while preserve the details of the object function $y(x)$.

We denote the noisy version of $y_h(x)$ as $\hat{y}_h(x)$. The main intention in the adaptive estimation is to select h in $\hat{y}_h(x)$ in such a way that the estimate is close to $y(x)$ as much as possible.

Let us start from derivation of the equation for the estimate $\hat{y}_h(x)$. Applying the kernel operator g_h to the both sides of the equation (5.1) we have

$$\begin{aligned} z_h(x) = (g_h \circledast (y \circledast v))(x) + (g_h \circledast \varepsilon)(x) = \qquad (5.13) \\ (v \circledast (y \circledast g_h))(x) + \varepsilon_h(x) = \\ (y_h \circledast v)(x) + \varepsilon_h(x) = (y \circledast v_h)(x) + \varepsilon_h(x), \qquad (5.14) \end{aligned}$$

where

$$\begin{aligned} & z_h(x) = (g_h \circledast z)(x), \\ & y_h \triangleq y_h(x) = (g_h \circledast y)(x), \\ & v_h \triangleq v_h(x) = (g_h \circledast v)(x), \\ & \varepsilon_h(x) = (g_h \circledast \varepsilon)(x). \end{aligned}$$

In the frequency domain these equations can be presented in two equivalent forms:

$$\begin{aligned} Z_h(f) = Y_h(f)V(f) + \varepsilon_h(f), \qquad (5.15) \\ Z_h(f) = Y(f)V_h(f) + \varepsilon_h(f), \end{aligned}$$

where $Z_h(f)$, $Y_h(f), V_h(f)$ and $\varepsilon_h(f)$ stay for the $2D$ DFT of the corresponding functions $z_h(x)$, $y_h(x)$, $v_h(x)$, and $\varepsilon_h(x)$.

It is clear that $Z_h(f) = G_h(f)Z(f)$, $Y_h(f) = G_h(f)Y(f)$ as well as $V_h(f) = G_h(f)V(f)$, where $G_h(f)$ is the $2D$ DFT of $g_h(x)$.

We use the following three types of "solutions" for (5.15) as estimates of $y(x)$:

(A) *Pure inversion (PI)*

$$\hat{Y}_h(f) = \frac{1}{V(f)} Z_h(f), \; V(f) \neq 0, \, f \in F, \tag{5.16}$$

(B) *Regularized pure inversion (RI)*

$$\hat{Y}_h(f) = \frac{V(-f)}{|V(f)|^2 + \varepsilon^2} Z_h(f), \tag{5.17}$$

(C) *Regularized Wiener inversion (RWI)*

$$\begin{gathered}\hat{Y}_h(f) = \frac{V(-f)|Y_h(f)|^2}{|V(f)Y_h(f)|^2 + \varepsilon^2\sigma^2|G_h(f)|^2} Z_h(f) = \\ \frac{V(-f)|G_h(f)|^2|Y(f)|^2}{|G_h(f)|^2|V(f)Y(f)|^2 + \varepsilon^2\sigma^2|G_h(f)|^2} Z_h(f) = \\ \frac{V(-f)|Y(f)|^2}{|V(f)Y(f)|^2 + \varepsilon^2\sigma^2} Z_h(f).\end{gathered} \tag{5.18}$$

Here $\sigma^2|G_h(f)|^2$ is the spectrum of the random noise at the output of the LPA kernel filter and $|Y(f)|^2$ stay for the spectrum of the image to estimate. The hat on $\hat{Y}_h(f)$ stays for estimate.

The parameter ε, called the regularization parameter, controls the trade-off between the amount of a signal distortion (smoothing) and of noise suppression. We believe that in what follows the regularization parameter ε cannot be confused with the noise $\varepsilon(x)$.

The pure Wiener inversion corresponds to $\varepsilon = 1$. The inversion 5.18 is the most general from the above three algorithms. If $\varepsilon = 0$, then (5.18) gives the PI (5.16). If the spectrum $|Y(f)|^2$ being the weight in the Wiener filter is assumed to be equal to 1, then (5.18) coincides with the RI (5.17).

The pure inversion (5.16) is a direct implementation of the basic idea. The estimate $\hat{y}_h(x)$ defined as the inverse Fourier transform of $\hat{Y}_h(f)$ is unbiased with respect to the smoothed object $y_h(x)$ with the variance $\sigma_y^2(x,h)$:

$$E\{\hat{y}_h(x)\} = y_h(x), \tag{5.19}$$

$$\sigma_y^2(x,h) = var\{\hat{y}_h(x)\} = \frac{\sigma^2}{n_1 n_2} ||\frac{G_h(f)}{V(f)}||^2, \tag{5.20}$$

where the Euclidian norm means $||\frac{G_h(f)}{V(f)}||^2 = \sum_{f \in F} |\frac{G_h(f)}{V(f)}|^2$.

The properties of the kernel operator g_h should be agreed with the PSF

convolution kernel v. Roughly speaking the ratio $G_h(f)/V(f)$ and the estimate variance $\sigma_y^2(x,h)$ should be finite and a decaying function of h as h is growing. The low pass filter operator g_h should be stronger than the PSF operator v in order to suppress high-frequency components in the output of the inverse PSF operator.

In Section 5.3 we present accurate statements concerning the asymptotic accuracy of the PI estimate for two classes of the $PSFs$ and formulate restrictions on the join properties of the convolution and estimate operators.

Actually, the kernel transfer function $G_h(f)$ is able to transform the initial problem, ill-posed with respect to the object $Y(f)$, in a well-defined with the respect to the smoothed object $Y_h(f)$ or at least to improve the conditioning of the initial problem.

The regularized inversion RI (5.17) is used instead of PI when $V(f)$ can be equal to zero at some f or in order to improve the conditioning of the inverse operator $1/V(f)$.

In the regularized Wiener inversion RWI (5.18) the algorithm requires knowledge of the object function $y(x)$ as the transfer function of the Wiener filter depends on $|Y(f)|^2$. An independent estimate of $|Y(f)|^2$ can be used in (5.18), for instance as it is implemented in [100], or the algorithm can be recursive and uses in (5.18) the estimate of $Y_h(f)$ obtained in the pervious steps of the procedure.

5.2.2 Pointwise adaptive deblurring

Further, the parameter h is selected in such way that the noise in $\hat{y}_h(x)$ will be suppressed as much as possible provided that the specific features of the object $y(x)$ are preserved in $\hat{y}_h(x)$. The ICI rule is used in order to achieve this goal in the pointwise manner, i.e. to find the adaptive varying window size for every x.

Let H be a finite set of smoothing parameter h:

$$H = \{h_1 < h_2 < .... < h_J\}, \tag{5.21}$$

starting with a quite small h_1, and determine a sequence of the confidence intervals $D(j)$ (2.5) of the biased estimates obtained with the windows $h = h_i$ as follows

$$D(i) = [\hat{y}_{h_i}(x) - \Gamma \cdot \sigma_y(x, h_i),\ \hat{y}_{h_i}(x) + \Gamma \cdot \sigma_y(x, h_i)], \tag{5.22}$$

where $\sigma_y(x, h_i)$ is the standard deviation of the estimate $\hat{y}_{h_i}(x)$ and Γ is a threshold (2.4) of the confidence interval.

The standard ICI rule (2.7) is used in order to obtain the adaptive window size. Determine the sequence of the upper and lower bounds of the confidence intervals $D(i)$ as follows

$$\begin{aligned} &D(i) = [L_i,\ U_i], \\ &U_i = \hat{y}_{h_i}(x) + \Gamma \cdot \sigma_y(x, h_i), \\ &L_i = \hat{y}_{h_i}(x) - \Gamma \cdot \sigma_y(x, h_i). \end{aligned} \tag{5.23}$$

Let

$$\begin{aligned} &\bar{L}_{i+1} = \max[\bar{L}_i,\ L_{i+1}],\ \underline{U}_{i+1} = \min[\underline{U}_i,\ U_{i+1}], \\ &i = 1, 2, ..., J,\ \bar{L}_1 = L_1,\ \underline{U}_1 = U_1 \end{aligned} \tag{5.24}$$

then the adaptive window length h_i^+ is defined by the largest i when

$$\bar{L}_i \leq \underline{U}_i \tag{5.25}$$

is still satisfied. This i^+ is the largest of those i for which the confidence intervals $D(i)$ have a point in common as it is discussed above in Section (2.3). It defines the adaptive window size and the adaptive LPA estimate as follows

$$\hat{y}^+(x) = \hat{y}_{h^+(x)}(x),\ h^+(x) = h_{i^+}. \tag{5.26}$$

We wish to emphasize once more that this window size ICI selection procedure requires a knowledge of the estimate and its variance only. It is equally applicable to all three introduced algorithms PI, RI, RWI. The only difference between the algorithms is that the corresponding estimates $\hat{y}_h(x)$ are given by (5.16), (5.17), (5.18) and their variances defined respectively by the formulas:

$$\sigma_y^2(x, h) = \frac{\sigma^2}{n_1 n_2} || \frac{G_h(f)}{V(f)} ||^2, \tag{5.27}$$

$$\sigma_y^2(x, h) = \frac{\sigma^2}{n_1 n_2} || \frac{V(-f) G_h(f)}{|V(f)|^2 + \varepsilon^2} ||^2, \tag{5.28}$$

$$\sigma_y^2(x, h) = \frac{\sigma^2}{n_1 n_2} || \frac{V(-f) |Y(f)|^2 G_h(f)}{|V(f) Y(f)|^2 + \varepsilon^2 \sigma^2} ||^2. \tag{5.29}$$

The threshold Γ in (5.23) is a design parameter of the algorithms.

5.2.3 Optimality of pointwise adaptive deblurring

The theoretical analysis produced in [40] for $1D$ deblurring shows that the ICI adaptive window size selection gives the best possible pointwise MSE and in this way it proves that the adaptive window sizes are close to the idea best possible choice of h assuming that the the object function $y(x)$ is known. This claim can be extended to the considered $2D$ discrete observation deblurring imaging.

5.3 Accuracy of LPA deblurring

Different filter design methods can be used for g_h where h is just one of the parameters responsible for the bandwidth of the filter. We find that the LPA is one of the useful technique here because it is natural way to obtain the filters obeying to the vanishing moment equations (5.12) and the local approximation idea perfectly fits to the image restoration problem.

What follows assumes that the LPA kernels g_h are applied for deblurring. The accuracy analysis produced in this section is based on the asymptotic integral representation of the LPA estimates and the corresponding accuracy results presented in Section (1.7.2).

5.3.1 Useful asymptotics

Let the object $y(x)$ can be treated as a $2D$ function defined on the field of continuous argument and the observations be sampled values of this function obtain on grid with the sampling interval Δ on the both variables x_1 and x_2. Then, provided that $\Delta \rightarrow 0$, $h/\Delta \rightarrow \infty$ the introduced LPA operator $g_h(x)$ allows the integral representation 1.6.

This representation is beneficial at least in two aspects. Firstly, it provides a clear links with integral smoothing LPA operators commonly used for the filter design and in many applications. Secondly, this integral representation gives a simple and explicit dependence of the operator on h which is useful for understanding of the nature of this operators as well as for accuracy analysis. Once more we wish to note the meaning of the assumption $h/\Delta \rightarrow \infty$. The parameter h is a "size" of the operator window and the ratio h/Δ is a number of samples in this window. The large number of samples enables one to replace the convolution-sum of the discrete operator by the corresponding convolution integral.

In the frequency domain the DFT is replaced by the integral Fourier transform (IFT) with the definition

$$G(\lambda) = \int_{R^2} g(x) \exp(-i2\pi < \lambda, x >) dx, \tag{5.30}$$
$$g(x) = \int_{R^2} G(f) \exp(i2\pi < \lambda, x >) d\lambda.$$

Here $< \lambda, x >= \lambda_1 x_1 + \lambda_2 x_2$ is the inner product of λ and x, and λ is a notation for the frequency in the integral Fourier transform.

Remind that the Parseval's equation in the integral case is of the form

$$\int_{R^2} g^2(x) dx = \int_{R^2} |G(\lambda)|^2 d\lambda. \tag{5.31}$$

The sampling interval Δ means that the lattice X in (5.1) is replaced by $X\Delta$. Then, the discrete convolution in the observation model (5.1) has a sense for $\Delta \to 0$ only provided that the kernel v is replaced by Δv.

Then

$$(y \circledast v)(x) \to \int_{R^2} v(u)y(x-u)du.$$

In all limit considerations it is assume that v is replaced by Δv.

Proposition 7 *: Let the window w in (1.29) be a continues function. Then the following hold as $\Delta \to 0$ and $h/\Delta \to \infty$:*

(1) The smoothed object function $y_h(x)$ approaches to the integral representation

$$y_h(x) \to \frac{1}{h^2}\int_{R^2} g(\frac{x-u}{h})y(u)du = \int_{R^2} g(u)y(x-hu)du, \tag{5.32}$$

where

$$\begin{aligned} g(x) &= w(x)\phi^T(0)\Phi^{-1}\phi(x), \\ \Phi &= \int_{R^2} w(u)\phi(u)\phi^T(u)du, \end{aligned} \tag{5.33}$$

(2) The DFT $G_h(f)$ of the estimator kernel $g_h(u)$ approaches to the form

$$G_h(f) \to \frac{1}{h^2}\int_{R^2} g(\frac{x}{h})\exp(-i2\pi < \lambda, x >)dx = G(\lambda h), \tag{5.34}$$

where $G(\lambda)$ is defined by (5.30),

(3) For the squared norm of $g_h(x)$ we obtain

$$\sum_x g_h^2(x) \to \frac{\Delta^2}{h^2}\int_{R^2} g^2(u)du = \Delta^2 \int_{R^2} |G(\lambda h)|^2 d\lambda. \tag{5.35}$$

(4) The formulas (5.27)-(5.29) for the variances approaches to the form

$$\sigma_y^2(x,h) \to \sigma^2\Delta^2 \int |\frac{G_h(\lambda)}{V(\lambda)}|^2 d\lambda, \tag{5.36}$$

$$\sigma_y^2(x,h) \to \sigma^2\Delta^2 \int |\frac{V(-\lambda)G_h(\lambda)}{|V(\lambda)|^2+\varepsilon^2}|^2 d\lambda, \tag{5.37}$$

$$\sigma_y^2(x,h) \to \sigma^2\Delta^2 \int |\frac{V(-\lambda)|Y(f\lambda)|^2 G_h(\lambda)}{|V(\lambda)Y(\lambda)|^2+\varepsilon^2}|^2 d\lambda. \tag{5.38}$$

The proof of Proposition 7 is given in Section 5.6.1 .

5.3.2 Mean squared error of PI

Let us consider the accuracy analyses of the PI as an estimator of the object $y(x)$. The estimation error is defined as a difference between the estimate and the estimated signal $e_y = y(x) - \hat{y}_h(x)$. For the observation model (5.1) and the estimator (5.16) this errors are composed from the systematic (bias), $E\{e_y\}$, and the random component, $e_y^0 = e_y - E\{e_y\}$.

The expectation of $\hat{y}_h(x)$ is equal to $y_h(x)$ and thus

$$E\{e_y\} = y(x) - y_h(x). \tag{5.39}$$

The variance of the random component in the frequency domain is calculated according to (5.27).

We produce the accuracy results for the asymptotic case $h/\Delta \to \infty$, where $\Delta \to 0$ is the sampling interval of the lattice X equal for both variables. As it was mentioned above this assumption mainly means that the large number of observations is involved in the kernel operator estimation and the summation in the discrete convolution can be replaced for the analysis purposes by the corresponding $2D$ integrals.

Actually this assumption is introduced mainly on technical reasons in order to obtain the bias and the variance in the form where the dependence on h is given explicitly. It allows to find the locally optimal values of h and give a clear presentation of the ICI adaptive scale algorithm.

Further the following is used in the analysis:

(A1) The object function $y(x)$ is deterministic locally smooth belonging to a nonparametric class of continuous $r-$differentiable functions (1.4)

$$F_r(\bar{L}_r) = \{\max_{r_1+r_2=r} \left| D^{(r_1,r_2)} y(x) \right| = L_r(x) \leq \bar{L}_r, \ \forall \ r_1 + r_2 = r\}, \tag{5.40}$$

where $D^{(r_1,r_2)} = \frac{\partial^{r_1+r_2}}{\partial x_1^{r_1} \partial x_2^{r_2}}$ is a differentiation operator and $\bar{L}_r$ is a finite constant.

(A2) The discrete convolution kernel in (5.1) is a sampled continues function $v(x)$ with the IFT $V(\lambda)$ satisfying one of two conditions:

(a) Polynomial decaying Fourier transform (ordinary smooth PSF).

There exist real positive c_0, c_1, A, α such that

$$\begin{gathered} |V(\lambda)| \geq c_0 ||\lambda||^{-\alpha}, \ \forall \ ||\lambda|| > A, \\ \min_{||\lambda|| \leq A} |V(\lambda)| = c_1, \end{gathered} \tag{5.41}$$

(b) Exponentially decaying Fourier transform (supersmooth PSF).

There exist real positive c_0, c_1, A, α such that

$$|V(\lambda)| \geq c_0 \exp(-\gamma||\lambda||^{\alpha}),\ \forall\ ||\lambda|| > A, \qquad (5.42)$$
$$\min_{||\lambda|| \leq A} |V(\lambda)| = c_1.$$

(A3) The estimator kernel g defined by (5.33) has IFT $G(\lambda)$ (5.30) satisfying one of the conditions:

(a)

$$\int_{R^2} |G(\lambda)|^2 d\lambda = B_0^2 < \infty, \qquad (5.43)$$
$$\int_{R^2} |G(\lambda)|^2 ||\lambda||^{2\alpha} d\lambda = B_1^2 < \infty, \quad \int_{R^2} |G(\lambda)| \cdot ||\lambda||^{\alpha} d\lambda < \infty,$$

(b) $G(\lambda)$ is supported on the finite rectangular $|\lambda_1| \leq \bar{\Lambda},\ |\lambda_2| \leq \bar{\Lambda}$ and

$$\int_{R^2} |G(\lambda)|^2 d\lambda = B_0^2 < \infty. \qquad (5.44)$$

The conditions (A2) give the lower bound on the rate of decaying $|V(\lambda)|$ as $||\lambda|| \to \infty$. They are given in the form of restrictions on frequency properties of the convolution kernel v. The conditions (5.41) and (5.42) define ordinary smooth and supersmooth PSF, respectively.

The conditions (A3) give the properties of the filter g_h as they are agreed with the properties of $V(\lambda)$.

We use the kernel estimate g_h subjected to the assumptions (A3 a) for deblurring of the ordinary smooth PSF and g_h subjected to (A3 b) for deblurring of the supersmooth PSF. This combination of the assumptions (A2 a)-(A2 b) with (A3 a)-(A3 b) suffices the bounded variance of the output signal of the system $G(\lambda)/V(\lambda)$ and the considered estimates for all $h > 0$.

The results are different for the ordinary and super smooth PSF and we present them separately.

5.3.3 Ordinary smooth PSF

Proposition 8 : *Consider the discrete PI estimate (5.16) with g_h of the order m given by (1.29) for the asymptotic $\Delta \to 0$, $h/\Delta \to \infty$. Let the assumptions (A1), (A2 a) and (A3 a) hold. Then:*

(I) The estimation bias is bounded as

$$|E\{e_y\}| \leq m_e(x,h), \qquad (5.45)$$
$$m_e(x,h) = h^M L_M(x) A_e,$$

where

$$M = min(m+1, r), \qquad (5.46)$$

and

$$A_e = \sum_{|r|=M} \frac{1}{r!} \int_{R^2} |g(u)u^r| du; \tag{5.47}$$

(II) The estimation variance is bounded as

$$\begin{aligned} & var\{e_y^0\} \leq \sigma_y^2(x,h), \\ & \sigma_y^2(x,h) = \sigma^2(\frac{1}{c_1^2}B_0^2\frac{\Delta^2}{h^2} + \frac{1}{c_0^2}B_1^2\frac{\Delta^2}{h^{2\alpha+2}}). \end{aligned} \tag{5.48}$$

The proof of this proposition is given in Section 5.6.2.

Note that in (5.47) r is the multi-index, i.e. $r! = r_1!r_2!$, $u^r = u_1^{r_1}u_2^{r_2}$, $|r| = r_1 + r_2$.

Here $m_e(x,h)$ in (5.45) and $\sigma_y^2(x,h)$ in (5.48) denote the upper bounds of the bias and the variance, respectively.

The finite variance of the estimate (5.48) means that the estimate (5.16) is well-defined for any $h > 0$.

It follows from (5.45) and (5.48) that the upper bound of the pointwise mean squared risk $r(x,h)$ of $y(x)$ estimation can be represented as follows

$$\begin{aligned} & r(x,h) \triangleq E(y(x) - \hat{y}_h(x))^2 \leq \bar{r}(x,h), \\ & \bar{r}(x,h) = m_e^2(x,h) + \sigma_y^2(x,h) = \\ & (h^M L_M(x)A_e)^2 + \sigma^2\frac{\Delta^2}{h^{2\alpha+2}}(\frac{1}{c_1^2}B_0^2h^{2\alpha} + \frac{1}{c_0^2}B_1^2). \end{aligned} \tag{5.49}$$

This upper bound risk $\bar{r}(x,h)$ is convex on h. Its minimization on h gives an "ideal" scale

$$h^* = \arg\min_h \bar{r}(x,h),$$

found from the equation

$$\frac{\partial}{\partial h}\bar{r}(x,h) = 0.$$

Let us assume that $\frac{1}{c_1^2}B_0^2h^{2\alpha} << \frac{1}{c_0^2}B_1^2$. Then the formula for the variance in (5.49) is simplified to

$$\sigma_y^2(x,h) \simeq \tilde{\sigma}_y^2(x,h) = \sigma^2\Delta^2B_1^2/(c_0^2h^{2\alpha+2})$$

and an explicit analysis can be produced.

Calculations show that in this case

$$h^*(x) = \left(\frac{\sigma^2B_1^2\Delta^2}{c_0^2A_e^2L_M^2(x)}\gamma^2\right)^{1/(2M+2\alpha+2)}, \ \gamma^2 = \frac{\alpha+1}{M} \tag{5.50}$$

and

$$\bar{r}^*(x) = \bar{r}(x, h^*(x)) = \sigma_y^2(x, h^*(x))(1+\gamma^2), \tag{5.51}$$
$$\gamma = m_e(x, h^*(x))/\tilde{\sigma}_y(x, h^*(x)),$$
$$\tilde{\sigma}_y^2(x, h^*(x)) = \sigma^2 B_1^2 \frac{\Delta^2}{c_0^2(h^*(x))^{2\alpha+2}}, \tag{5.52}$$
$$\gamma = \sqrt{\frac{\alpha+1}{M}} \tag{5.53}$$

Here the parameter γ is introduced as a ratio of the bias $m_e(x, h^*(x))$ to the estimation standard deviation $\tilde{\sigma}_y(x, h^*(x))$, when the both are calculated for the ideal scale $h^*(x)$. These calculations show that this ratio is a constant depending only on $M = \min\{m+1, r\}$ and the properties of the convolution kernel given by the parameter α.

Thus, γ depends only on the power LPA, the smoothness of the function defined by the value of r and the filtering properties of the convolution kernel. This result is exactly similar to discussed in Section (1.7.3).

The formulas (5.50) and (5.51) demonstrate that the ideal scale $h^*(x)$ depends on the $M-th$ derivatives of $y(x)$ and the ideal variance-bias trade-off is achieved when the ratio between the bias and standard deviation γ is equal to $\sqrt{\frac{\alpha+1}{M}}$. It can be seen also that

$$|m_e(x,h)| = \begin{cases} < \gamma \cdot \sigma_y(x,h) \text{ if } h < h^*(x) \\ > \gamma \cdot \sigma_y(x,h) \text{ if } h > h^*(x). \end{cases}. \tag{5.54}$$

Test of the hypotheses: $h \lesseqgtr h^*(x)$ is used in the ICI rule in order to obtain the adaptive values of the scale approximating $h^*(x)$.

Inserting (5.50) in (5.52) we can see that the order of the MSE with respect to Δ is defined as

$$\sigma_y^2(x, h^*(x)) = \mathit{0}(\Delta^{\frac{2M}{M+\alpha+1}}). \tag{5.55}$$

The sampling interval Δ in (5.1) is inverse proportional to the numbers of samples on x_1 and x_2. Let $n_1 = n_2 = n$. It means that with respect to n (5.55) gives

$$\sigma_y^2(x, h^*(x)) = \mathit{0}(n^{-\frac{2M}{M+\alpha+1}}). \tag{5.56}$$

This formula is quite interesting. It shows the ideal MSE convergence rate with respect to the number of samples in the image. It is the best possible convergence rate achieved provided that the optimal scale is selected, which actually requires the knowledge of the derivatives of $y(x)$. This convergence rate cannot be better than $\mathit{0}(n^{-2})$. Increasing M, i.e. the smoothness of $y(x)$ and the LPA order m, improves the convergence rate, while the increasing α results in the slower convergence rate. With $\alpha = 0$

the formula (5.56) gives the results coinciding with ones obtained for the nonparametric estimation in the direct observations which do not require the data deblurring (see Section 1.7).

The deblurring reduces the ideal convergence rate.

5.3.4 Supersmooth PSF

Let start from a simple example. Assume that

$$g(x) = \frac{1}{2\pi\sigma_g^2}\exp(-\frac{||x||^2}{2\sigma_g^2}).$$

This kernel because of its symmetry defines the kernel estimator of the first order. In the frequency domain it corresponds to

$$G(\lambda) = \exp(-2\sigma_g^2||\lambda||^2\pi^2).$$

Assume also that a supersmooth PSF is given as the $2D$ Gaussian density

$$v(x) = \frac{1}{2\pi\sigma_v^2}\exp(-\frac{||x||^2}{2\sigma_v^2})$$

with the IFT as

$$V(\lambda) = \exp(-2\sigma_v^2||\lambda||^2\pi^2).$$

We have for the variance of estimation (5.36)

$$\int_{R^2}|\frac{G(h\lambda)}{V(\lambda)}|^2 d\lambda = \int_{R^2}\exp|\frac{G_h(\lambda)}{V(\lambda)}|^2 d\lambda =$$
$$\int_{R^2}\exp((-h^2\sigma_g^2+\sigma_V^2)||\lambda||^2\pi^2 2)d\lambda =$$
$$= \begin{cases} \frac{1}{2\pi(h^2\sigma_g^2-\sigma_V^2)}, \text{ if } h^2\sigma_g^2-\sigma_V^2>0 \\ \infty, \text{ otherwise} \end{cases}.$$

The estimation variance is finite only provided that $h > \sigma_v/\sigma_g$. Thus, in this case it is not allowed to use small h and only the small h enable the estimator to preserve the image details.

It is an example what can happen if there is no a proper coordination between the properties of the estimator g and PSF v. In order to obtain a finite variance for all h for the super smooth PSF the kernel estimator should be a stronger filter in the high frequency domain.

The finite frequency support assumption in (A3 b) is an example of such stronger filter used in the following proposition.

However, even we achieve the finite variances for all h for the super smooth PSF the asymptotic convergence rate appeared to be much slower than it is for the ordinary smooth PSF. It give an illustration of the general claim that the deblurring of the supersmooth PSF is more difficult problem.

Proposition 9 : *Consider the discrete PI estimate (5.16) with g_h of the order m given by (1.29) for the asymptotic $\Delta \to 0$, $h/\Delta \to \infty$. Let the assumptions (A1), (A2 b) and (A3 b) hold. Then*

(I) The estimation bias is bounded as in (5.45);

(II) The estimation variance is bounded as

$$var\{e_y^0\} \leq \begin{cases} \sigma^2 B_0^2 \Delta^2 (\frac{1}{c_1^2 h^2} + \frac{\exp(2\gamma\bar{\Lambda}^\alpha h^{-\alpha})}{c_0^2 h^2}), \ \textit{if}\ h \leq A/\bar{\Lambda}, \\ \sigma^2 B_0^2 \Delta^2 \frac{1}{c_1^2 h^2}, \ \textit{if}\ h > A/\bar{\Lambda}. \end{cases} \tag{5.57}$$

(III) For the window size $h = c(\log \frac{1}{\Delta^2})^{-1/\alpha}$ with $c > (2\gamma)^{1/\alpha}\bar{\Lambda}$ the mean square error of the estimation bounded as

$$r(x,h) \leq (cL_M(x)A_e)^2((\log \frac{1}{\Delta^2})^{-2M/\alpha} + o(1)), \tag{5.58}$$

where $o(1)$ stays for terms smaller in order.

The proof of Proposition 9 is given in Section 5.6.3.

For $\Delta \sim 1/n$ the formula (5.58) gives

$$r(x,h) \leq (cL_M(x)A_e)^2((\log n^2)^{-2M/\alpha} + o(1)), \tag{5.59}$$

where n^2 is a total number of samples in the image.

Comparison (5.59) versus (5.56) shows that the convergence rate for deconvolution of super smooth PSF is much slower than that for the ordinary smooth PSF. Note also that there is no a better choice for the window $h = c(\log \frac{1}{\Delta^2})^{-1/\alpha}$ which is able to improve the asymptotic accuracy (5.59).

5.3.5 The ICI rule

The problem of the pointwise adaptive scale has been studied in Chapter 2. Here we apply this ICI technique. The above accuracy analysis is used to show that the ICI rule is applicable for the deblurring.

Ordinary smooth PSF

The ICI rule can be derived for the deconvolution problem from speculations which copy the logic exploited in Chapter 2 where the ICI rule was obtained for the direct observation.

According to the results given in Proposition 8 and formulas (5.49)-(5.53) the estimation error of the PI estimator can be represented in the form

$$|e(x,h)| = |y(x) - \hat{y}_h(x)| \leq \omega_e(x,h) + \left|e^0(x,h)\right|, \tag{5.60}$$

where $\omega_e(x,h)$ stays for the bias and $e^0(x,h) = y(x) - E\{\hat{y}(x,h)\}$ is a random error with the probability density $N(0, \sigma_y^2(x,h))$.

Using

$$\sigma_y^2(x,h) \leq \bar{\sigma}_y^2(x,h) = \sigma^2(B_0^2/h^2 + Q^2\Delta^2/h^{2\alpha+2}) \simeq \\ \bar{\sigma}_y^2(x,h) = \sigma^2 Q^2\Delta^2/h^{2\alpha+2}$$

we obtain that

$$\left|e^0(x,h)\right| \leq \chi_{1-\alpha/2} \cdot \bar{\sigma}_y(x,h)$$

holds with the probability $p \geq 1-\alpha$, where $\chi_{1-\alpha/2}$ is $(1-\alpha/2)-th$ quantile of the standard Gaussian distribution, and with the same probability p

$$|e(x,h)| \leq \omega_e(x,h) + \chi_{1-\alpha/2}\bar{\sigma}_y(x,h). \tag{5.61}$$

It follows from (5.54) that for $h \leq h^*(x)$ the inequality (5.61) can be weakened to

$$|e(x,h)| \leq \Gamma \cdot \bar{\sigma}_y(x,h), \ h \leq h^*(x), \tag{5.62}$$

$$\Gamma = \gamma + \chi_{1-\alpha/2}, \tag{5.63}$$

where according to (5.53)

$$\gamma = \sqrt{\frac{\alpha+1}{M}} = \sqrt{\frac{\alpha+1}{\min\{m+1, r\}}}.$$

Now we use a finite set of window sizes H (5.21) and, according to (5.62), determine a sequence of the confidence intervals $D(j)$ (5.22) of the biased estimates obtained with the windows $h = h_j$.

Then for $h = h_j$ (5.62) is of the form

$$y(x) \in D(j), \tag{5.64}$$

and we can conclude from (5.61) and (5.62) that while $h_j < h^*(x)$ holds for $h = h_j$, $1 \leq j \leq i$, all of the intervals $D(j)$, $1 \leq j \leq i$, have a point in common, namely, $y(x)$.

In the opposite case, when the intersection of the confidence intervals is empty it indicates that $h_j > h^*(x)$. Thus we arrive to the *ICI* rule in the form given in the section 2.2 with the adaptive scale estimate (5.26).

Super smooth *PSF*

It follows from Proposition 9 that the upper bounds of the bias and variance of the estimate are increasing and decreasing functions of h respectively. It immediately means that the justification of the *ICI* given in the previous section is completely valid for deconvolution of the super smooth PSF. The only difference is that it is impossible to obtain the accurate value of the optimal scale and to verify if the bias-to-variance is invariant as it is for the ordinary smooth PSF.

Thus, we can use the ICI scale selection for any ordinary and super smooth PSF.

5.4 Algorithm implementation

The implementation of the pointwise adaptive algorithms is completely similar to the $LPA - ICI$ procedures considered in Chapter 2.

Here we present the basic implementation of the PI, RI and WI methods. The practical algorithms include the basic ones as elements of more complex structures.

The basic PI algorithm consists of the following steps:

1. Set $H = \{h_1 < h_2 < .... < h_J\}$, m, Γ;

2. For $h = h_i$, $i = 1, ..., J$, calculate:

(a) The kernels $g_h(x)$,

(b) The DFT of the kernels $G_h(f)$,

(c) The variance of the estimates (5.27).

3. For $h = h_i$, $i = 1, ..., J$, and x_s, $s = 1, 2, .., N$, calculate the estimates $\hat{y}_h(x_s)$ according to the algorithm (5.16)

4. Use the ICI algorithm (5.23)-(5.25) to find the adaptive window sizes $h^+(x_s)$ for all x_s.

5. Define the adaptive estimates according to (5.26).

The step 2 defines a bank of the filters with different bandwidth h. Step 3 serves for calculation of the estimates for all $h \in H$. The ICI rule in step 4 selects the adaptive varying bandwidths for each x_s. The final step 5 gives the estimates with the adaptive bandwidths. Actually, the steps 4 and 5 are integrated in the ICI and produced simultaneously.

The basic RI algorithm consists of the same steps as the PI algorithm with the following two changes: the regularization parameter ω should be settled in Step 1 and the formula (5.28) is applied for variance calculation.

The basic WI algorithm consists of the same steps as the PI algorithm with the formula (5.29) is applied for variance calculation. It requires also knowledge of the object function which can be estimated by some auxiliary procedure independent on the main algorithm or obtained by the recursive implementation of the main algorithm using for the object function in (5.29) the estimates obtained on the previous steps.

The standard deviation σ used for the variance calculations in 2(a) is estimated according to (2.14).

A symmetric weight w is a good choice if $y(x)$ is isotropic in a neighborhood of the estimation point. However, if $y(x)$ is anisotropic directional varying, as it happens near discontinuities or image edges the nonsymmetric approximation of $y(x)$ is more reasonable way to obtain a better result. Two different approaches to deal with anisotropy of $y(x)$ are proposed. Those are the multiple window estimate in Section 2.6 and the combined window estimate in Section 2.7. These two approaches are completely applicable for nonparametric adaptive scale deblurring.

5.5 Simulation

The $LPA - ICI$ algorithms developed in Chapter 2 for image denoising and modified to the image deblurring problem in Section 5.4 are tested in a number of experiments.

The test signal is the 256 × 256 "Cameraman" image (8 bit gray-scale) corrupted by an additive zero-mean Gaussian noise. The blurred SNR ($BSNR$) is defined in dB as

$$BSNR = 10\log_{10}[\sum_{x} E\{(z(x) - (y \circledast v)(x))^2\}/\sigma^2 n_1 n_2]$$

with $n_1 = n_2 = 256$. The results presented are mainly obtained for $BSNR$=40dB. The discrete-space blur convolution PSF is 9 × 9 uniform box-car square .

For the parameters of the $LPA - ICI$ we assume: the power $m = 2$ and the Gaussian window

$$w(x) = \exp(-||x||^2/2\sigma_w^2)$$

with $\sigma_w = 0.4$.

The window size set (scales) used in the ICI is as in (3.4)

$$H = \{1, 2, 3, 4, 5, 7, 10, 14, 20, 29\}.$$

The basic design parameters of the proposed $LPA - ICI$ deblurring algorithm are the threshold Γ for the ICI and the regularization parameter ε for the deblurring operator. The bias-variance trade-off here is clear. Smaller values of these parameters result in decreasing smoothing properties of the algorithm, i.e. the distortion of the original image becomes smaller while the level of the random components becomes stronger. Larger values of the parameters Γ and ε result in a vise versa effects. For the $LPA - ICI$ algorithm this sort of behavior of the image denoising is shown, in particular, in [67] and it is a common knowledge for the regularization parameter.

For estimation of the original image spectrum $|Y(f)|^2$ we exploit the iterative Wiener algorithm proposed in [48].

First, we wish to determine the optimal or nearly optimal parameters Γ^* and ε^* from the analysis of the $LPA - ICI$ deblurring algorithm by optimizing performance criteria (3.7)-(3.12):

(1) Root mean squared error ($RMSE$):

$$RMSE = \sqrt{\frac{1}{\#}\sum_x (y(x) - \hat{y}(x))^2};$$

(2) SNR in dB:

$$SNR = 10\log_{10}\sum_x |y(x)|^2 / \sum_x |y(x) - \hat{y}(x)|^2;$$

(3) Improvement in SNR ($ISNR$) in dB:

$$ISNR = 20\log_{10}(\hat{\sigma}/RMSE);$$

(4) Peak signal-to-noise ratio ($PSNR$) in dB:

$$PSNR = 20 log_{10}(max_x|y(x)|/RMSE);$$

(5) Mean absolute error (MAE):

$$MAE = \frac{1}{\#}\sum_x |y(x) - \hat{y}(x)|;$$

(6) Maximum absolute difference (error):

$$MAXDIF = \max_x |y(x) - \hat{y}(x)|.$$

These criteria allow to evaluate the performance of the algorithm quantitatively, while $PSNR$ is treated as a criterion linked with a visual image perception. However, the visual evaluation is considered as an independent performance criterion required to be evaluated. For deblurring we use the multiple window version of the algorithms.

The performance criteria are calculated for symmetric and four quadrant window estimates as well as for the weighted final estimates (2.17). We have studied few versions of the fusion for the final estimate which are different in a window estimates used for fusion as well as the weights used in the fusion. In particular, we considered the final estimate obtained from four (quadrants 1,2,3,4 only) and from five estimates (quadrants 1,2,3,4 plus symmetric window). We considered also two types of the weights of the fusion: inverse variance of the estimates as in (2.17) and the simple mean of the fused estimates.

The analysis is in the favor of the four quadrant estimates (symmetric window estimate is not included) with the inverse variance weights what is used further. Sometimes the difference between these estimates is negligible. However, mainly the four-quadrant estimate gives a better detail preservation and better visual effects.

The above performance criteria achieve their extrema for slightly different values of Γ and ε. However, the criteria functions are relatively slow varying. It allows to select compromise values of Γ and ε which are nearly optimal for all criterion functions. These compromise values which can be treated as the rule of thumb for the parameter selection are $\Gamma^* = 1.6$ and $\varepsilon^* = 0.015$ for the RI algorithm and $\Gamma^* = 1.6$ and $\varepsilon^* = 0.4$ for the RWI algorithm. Naturally for the particular images and SNR the performance may be improved by further tuning and adaptation of these parameters.

For the considered image and the simulation scenario with box-car PSF and the Gaussian window for the LPA the behavior of the criteria as functions of Γ and ε for the symmetric and quadrant window estimates is illustrated in Figures 5.1 and 5.2 for the RI algorithm. The curves in these Figures show that the different criteria for different quadrant windowed estimates indeed can be agreed in terms of their extremum values achieved for the parameters Γ and ε having the compromise values shown above.

For the algorithm RWI similar curves are shown in Figures 5.3 and 5.4 with the similar conclusions on Γ and ε values.

Figure 5.5 shows the varying adaptive scales produced by the ICI rule in the RWI algorithm for each of the five (symmetric and nonsymmetric quadrant) windows. Small and large scales are shown by black and white, respectively. The adaptive scales are quite sensitive with respect to edges in images. The ICI rule selects smaller windows in a neighborhood of the intensity changes and larger window for areas of a flat image intensity. The adaptive window sizes delineate and shade contours of the cameraman, camera, and other image details. The obtained window sizes actually correspond to the intuitively clear behavior of the varying window size relevant to the smoothing of the data if the true image is known. The character of these figures is quite similar to having place in application of the ICI rule to image denoising of direct (not blurred) observations (see Chapter 3). The quadrant estimates corresponding to the ICI windows shown in Figure 5.5 can be seen in Figure 5.6. The final four quadrant fused estimate-images are of nearly equivalent quality for the RI and RWI methods.

One of these images the WRI image is shown in Figure 5.7b. For comparison purposes we show also: the blurred noisy image in Figure 5.7a and in Figure 5.7c,d the images restored by the image deblurring routines available from the Image Processing Toolbox (Version 3.1) of MATLAB.

Those are two routines deconvreg.m and deconvwnr.m implementing the regularized deconvolution and Wiener deconvolution, respectively.

Let us make clear details of using these routines. The regularized deconvolution was done in two stage manner as follows:

$$\begin{aligned}(1)\ [y1, LARGA] = &\quad deconvreg(z, PSF, n\sigma^2), \\ (2)\ [Matlab_1] = &\quad deconvreg(z, PSF, [], LARGA \cdot \varepsilon_1).\end{aligned} \tag{5.65}$$

Here z is the noisy blurred image, PSF is v, $LARGA$ is an auxiliary variable of the algorithm. The output image estimate is denoted as $Matlab_1$.

We use the parameter ε_1 in order to maximize the SNR of this estimate. The maximum was achieved at $\varepsilon_1 = 0.26$ with the optimal $SNR = 21.35$ dB and $ISNR = 6.16$ dB. The corresponding image restoration is given in Figure 5.7c.

The Wiener deconvolution routine has two modes of using different by information available on the noise. The first mode assumes that the NSR (noise-to-original image ratio) is known and we denote the corresponding image restoration as $Matlab_{21}$. The second mode assumes that the spectrums of the original image and the noise are given. Then, we denote the corresponding image restoration as $Matlab_{22}$.

The corresponding restorations are obtained using the routine as follows

$$\begin{aligned}(1)\ [Matlab_{21}] &= deconvwvr(z, PSF, NSR \cdot \varepsilon_2); \\ (2)\ [Matlab_{22}] &= deconvwvr(z, PSF, NCORR, ICORR),\end{aligned}$$

where $NCORR$, $ICORR$ are correlation matrices of the noise and the original image (details can be seen in MATLAB's Help).

We use the parameter ε_2 in the reconstruction $Matlab_{21}$ in order to maximize SNR of the corresponding estimate. The maximum was achieved at $\varepsilon_2 = 55$ and yields $SNR = 20.75$ dB and $ISNR = 5.567$ dB. This image estimate called $Matlab_{21}$ is demonstrated in Figure 5.7d.

Comparison of all three reconstructions is definitely in favor of the $LPA-ICI$ estimates as in the numerical values of the criteria as well as in visual evaluation. In particular, the both MATLAB reconstructions suffer from artifacts in the form of multiple spots spread over the image and in particular clear seen on the sky-background. The $LPA-ICI$ reconstruction gives also a sharper and cleaner image.

The reconstruction $Matlab_{22}$ gives a valuable improvement of the criteria values as much as $SNR = 22.07$ dB and $ISNR = 6.885$ dB. However, now for the fair competition we need to use the RWI reconstruction assuming that the spectrum of the original image in known exactly. In this case the the RWI algorithm also improves its performance. It gives the values $SNR = 22.70$ dB and $ISNR = 7.51$ dB as well as the visual quality is again in favor of the RWI algorithm.

This figures as well as visual evaluation demonstrate an advance performance of the developed algorithms.

In more details algorithm performance is characterized numerically by criteria values in Table 5.1.

These figures are obtained for the image intensity functions normalized to the segment [0,1]. The first two columns correspond to the developed $LPA-ICI$ estimates using the RWI and RI deblurring. Third-fifth columns shows results obtained by the algorithms of the Image Processing Toolbox. The all criteria figures are in favor of the $RWI-LPA-ICI$ algorithm, further follows the $RI-LPA-ICI$ algorithm and after that the algorithms $Matlab_1$, $Matlab_{21}$.

Table 5.1

Box-car blur PSF and Gaussian window LPA with $m = 2$

Criteria	RWI $\varepsilon = 0.4$	RWI $\varepsilon = 0.4$ $\|Y(f)\|$ given	RI $\varepsilon = 0.015$	$Matlab_1$ $\varepsilon_1 = 0.26$	$Matlab_{21}$ $\varepsilon_2 = 55$	$Matlab_{22}$ $\|Y(f)\|$ given
SNR, dB	22.04	22.70	21.94	21.35	20.75	22.07
$ISNR$,	6.855	7.515	6.749	6.162	5.567	6.88
$RMSE$, dB	0.0419	0.039	0.0424	0.0454	0.0486	0.0418
$PSNR$, dB	27.56	28.22	27.32	26.86	26.26	27.60
MAE	0.0249	0.0230	0.0251	0.0306	0.0331	0.0287
$MAX-DIFF$	0.4542	0.4337	0.4572	0.4920	0.5018	0.4450

Finally, Figure 5.8 shows the performance (SNR of the restored image) as a function of $BSNR$ for all considered algorithms. We use the design parameter values Γ, ε, ε_1 and ε_2 found from the analysis of the middle level $BSNR = 40$ dB. The RWI algorithm definitely demonstrates the best performance for all $BSNR$.

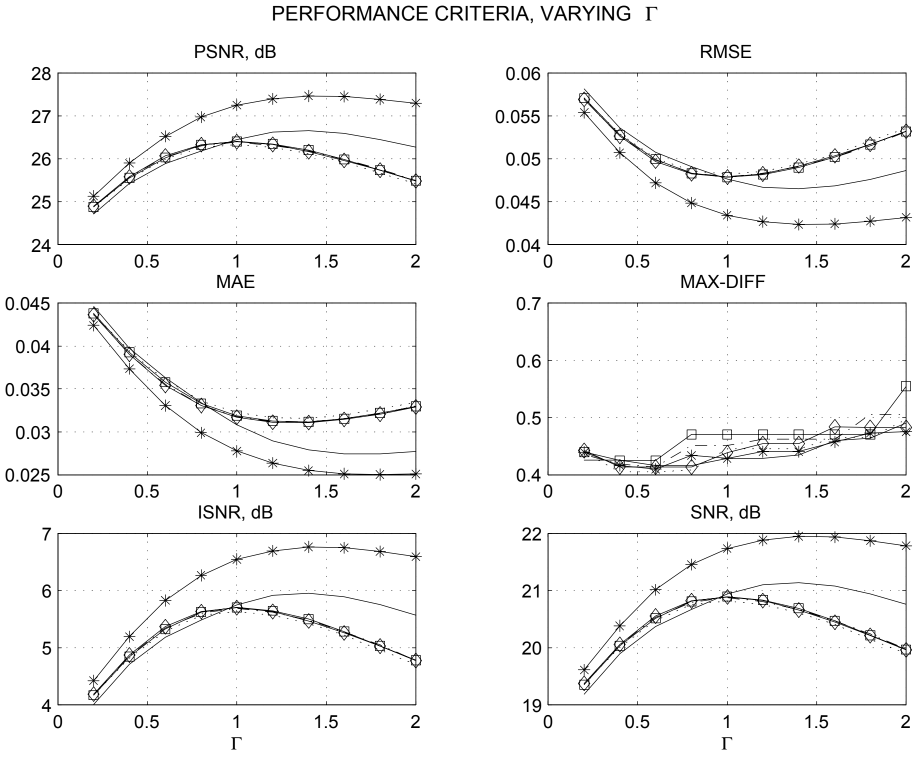


Figure 5.1: The Regularized Inverse (RI) $LPA-ICI$ estimates. The Car-Box PSF and Gaussian window of the LPA. The performance criteria as functions of the threshold Γ for $\varepsilon = 0.015$. Criteria are given for the symmetric, quadrant and final estimates. Notation used: solid line for the symmetric window estimate, stars * for the final fused estimate, dotted lines, dash-dot lines, lines with squares and diamonds for the Q_1, Q_2, Q_3, and Q_4 quadrant estimates respectively. Extremum values of SNR, $ISNR$, $PSNR$, $RMSE$ for the final estimate are at $\Gamma \simeq 1.6$. Symmetric and quadrant estimates achieve their extremum values at the values close to $\Gamma \simeq 1.6$. Thus, $\Gamma \simeq 1.6$ can be treated as a compromise value for all criteria and for all estimates.

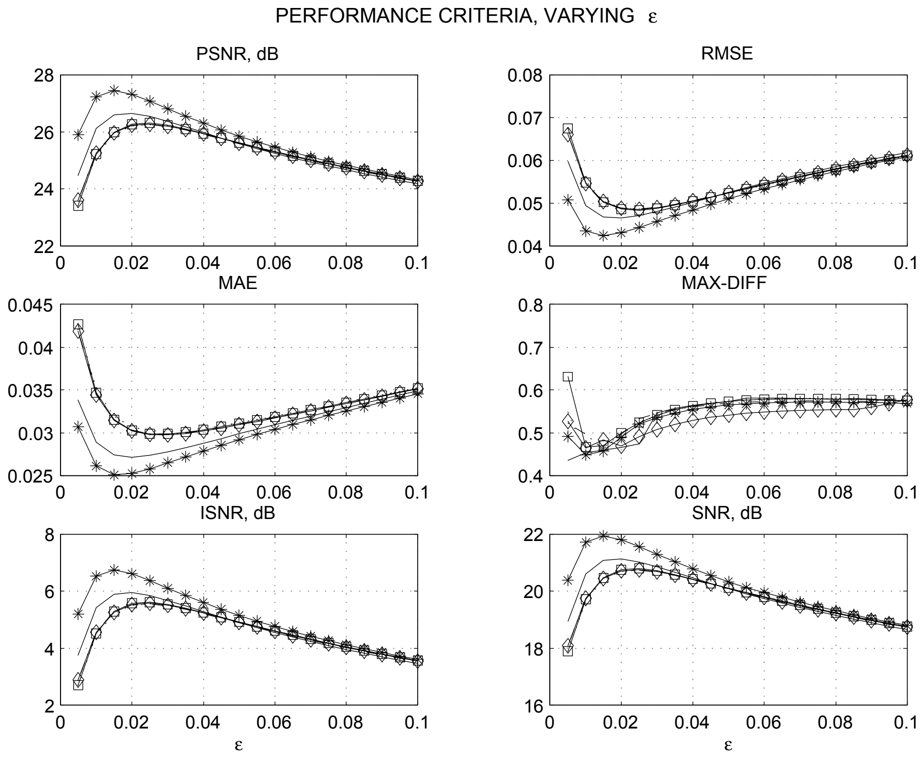


Figure 5.2: The Regularized Inverse (RI) $LPA - ICI$ estimates. The performance criteria as functions of the threshold ε for $\Gamma = 1.6$. Criteria are given for the symmetric, quadrant and final estimates. Notation used: solid line for the symmetric window estimate, stars * for the final fused estimate, dotted lines, dash-dot lines, lines with squares and diamonds for the Q_1, Q_2, Q_3, and Q_4 quadrant estimates respectively. All criteria have their extremum values for the final estimate at $\varepsilon \simeq 0.015$. Symmetric and quadrant estimates achieve their extremum values at ε values close to 0.015. Thus, $\varepsilon \simeq 0.015$ can be treated as a compromise optimal value for all criteria and for all estimates.

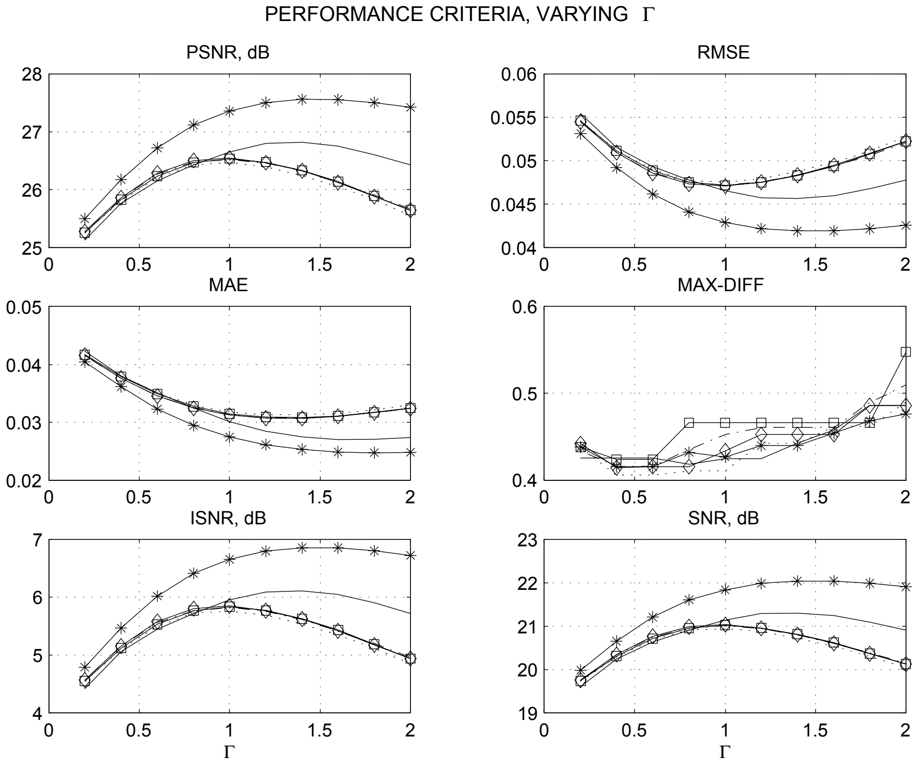


Figure 5.3: The Regularized Wiener Inverse (RWI) $LPA-ICI$ estimates. The Car-Box PSF and Gaussian window of the LPA. The performance criteria as functions of the threshold Γ for $\varepsilon = 0.4$. Criteria are given for the symmetric, quadrant and final estimates. Notation used: solid line for the symmetric window estimate, stars * for the final fused estimate, dotted lines, dash-dot lines, lines with squares and diamonds for the Q_1, Q_2, Q_3, and Q_4 quadrant estimates respectively. Extremum values of SNR, $ISNR$, $PSNR$, $RMSE$ for the final estimate are at $\Gamma \simeq 1.6$. Symmetric and quadrant estimates achieve their extremum values at the values close to $\Gamma \simeq 1.6$. Thus, $\Gamma \simeq 1.6$ can be treated as a compromise value for all criteria and for all estimates.

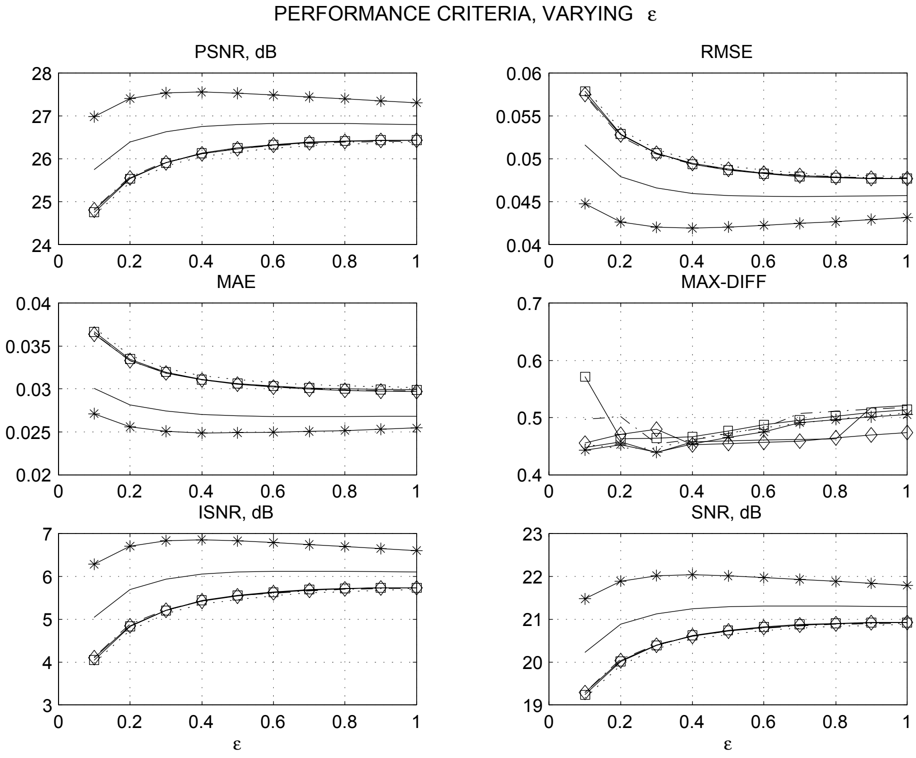


Figure 5.4: The Regularized Wiener Inverse (RWI) $LPA-ICI$ estimates. The performance criteria as functions of the threshold ε for $\Gamma = 1.6$. Criteria are given for the symmetric, quadrant and final estimates. Notation used: solid line for the symmetric window estimate, stars * for the final fused estimate, dotted lines, dash-dot lines, lines with squares and diamonds for the Q_1, Q_2, Q_3, and Q_4 quadrant estimates respectively. All criteria have their extremum values for the final estimate at $\varepsilon \simeq 0.4$. Symmetric and quadrant estimates achieve their extremum values at ε values close to 0.4. Thus, $\varepsilon \simeq 0.4$ can be treated as a compromise optimal value for all criteria and for all estimates.

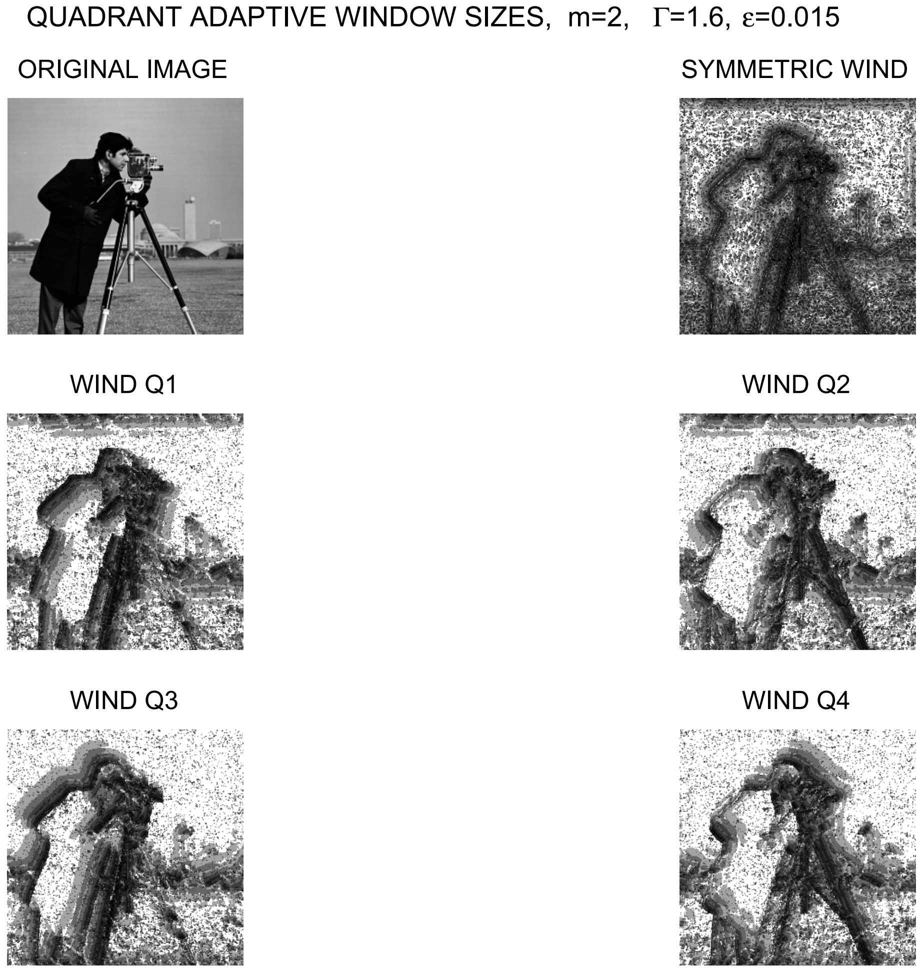


Figure 5.5: The RWI algorithm, Car-Box PSF, Gaussian window in the LPA. Cameraman images show true image and adaptive varying window sizes obtained by ICI for the symmetric window and quadrant image estimates. Small and large window sizes are shown by black and white, respectively. The adaptive window sizes are quite sensitive with respect to edges in images. The ICI rule selects smaller windows in a neighborhood of the intensity changes and larger window for areas of flat image intensity. The ICI sensitivity is directional and the varying window sizes are different for different quadrant estimates giving directional shadowing image edges. The adaptive window sizes delineate and shade contours of the cameraman, camera, and other image details. The obtained window sizes actually correspond to the intuitively clear behavior of the varying window size relevant to the smoothing of the data if the true image is known.

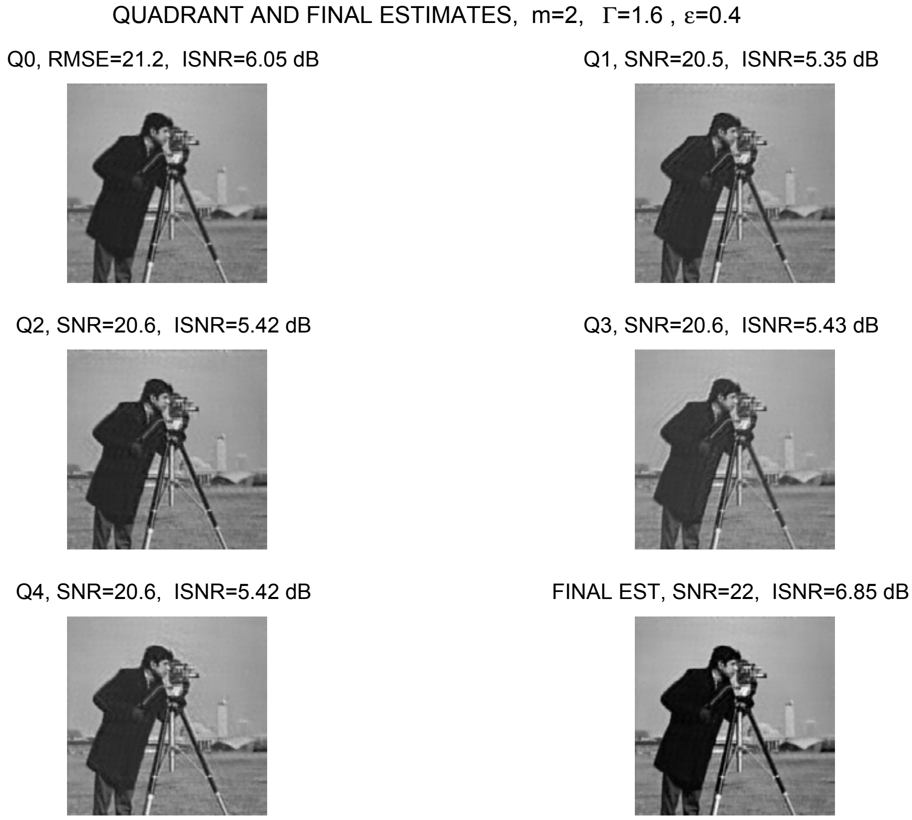

QUADRANT AND FINAL ESTIMATES, m=2, Γ=1.6 , ε=0.4
Q0, RMSE=21.2, ISNR=6.05 dB
Q1, SNR=20.5, ISNR=5.35 dB
Q2, SNR=20.6, ISNR=5.42 dB
Q3, SNR=20.6, ISNR=5.43 dB
Q4, SNR=20.6, ISNR=5.42 dB
FINAL EST, SNR=22, ISNR=6.85 dB

Figure 5.6: The RWI algorithm, Car-Box PSF, Gaussian window in the LPA. Quadrant and final four quadrant estimate.

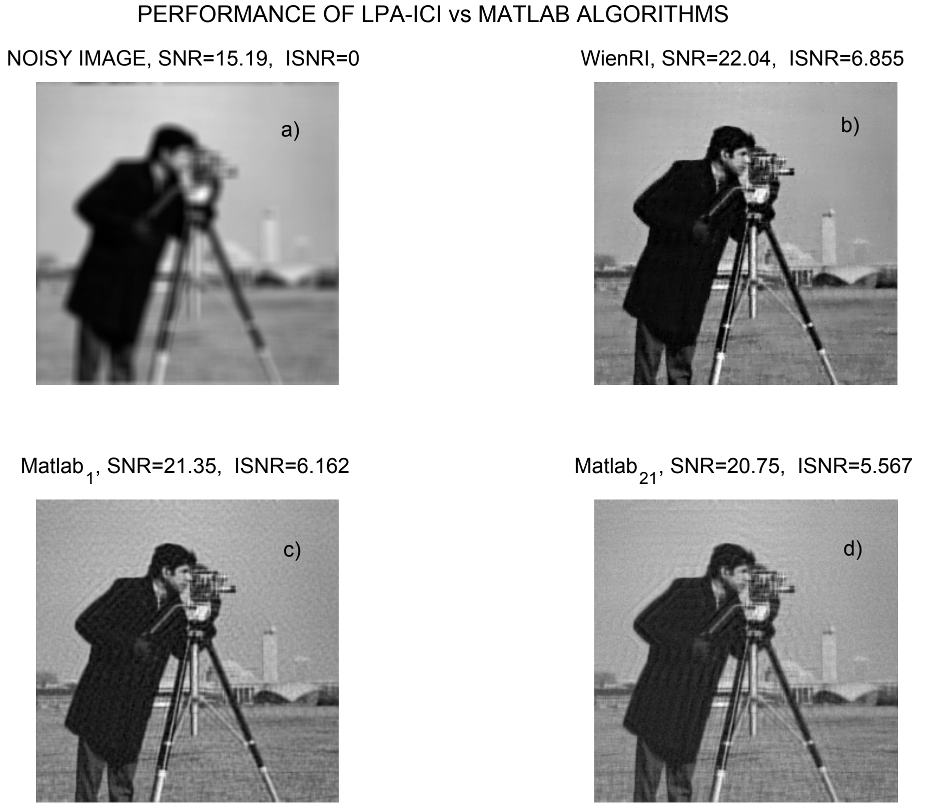

PERFORMANCE OF LPA-ICI vs MATLAB ALGORITHMS
NOISY IMAGE, SNR=15.19, ISNR=0
a)
WienRI, SNR=22.04, ISNR=6.855
b)
Matlab$_1$, SNR=21.35, ISNR=6.162
c)
Matlab$_{21}$, SNR=20.75, ISNR=5.567
d)

Figure 5.7: The RWI algorithm, Box-Car PSF, Gaussian window in the LPA. (a) The observed blurred noisy Cameraman image; (b) the final $LPA - ICI$ estimate fused from the four quadrant estimates. The estimate obtained by using the Wiener Inversion (RWI) demonstrates a performance with $SNR = 22$ dB and $ISNR = 6.85$ dB; (c) The estimate $Matlab_1$ obtain by the MATLAB routine deconvreg.m with optimization on regularization parameter with $SNR = 21.35$ dB and $ISNR = 6.16$ dB; (d) The estimate $Matlab_{21}$ obtain by the MATLAB routine deconvwnr.m with optimization on some regularization parameter with $SNR = 20.75$ dB and $ISNR = 5.57$ dB. The criteria values and the visual inspection of the image reconstructions are definitely in favor of the $LPA - ICI$ based RWI algorithm.

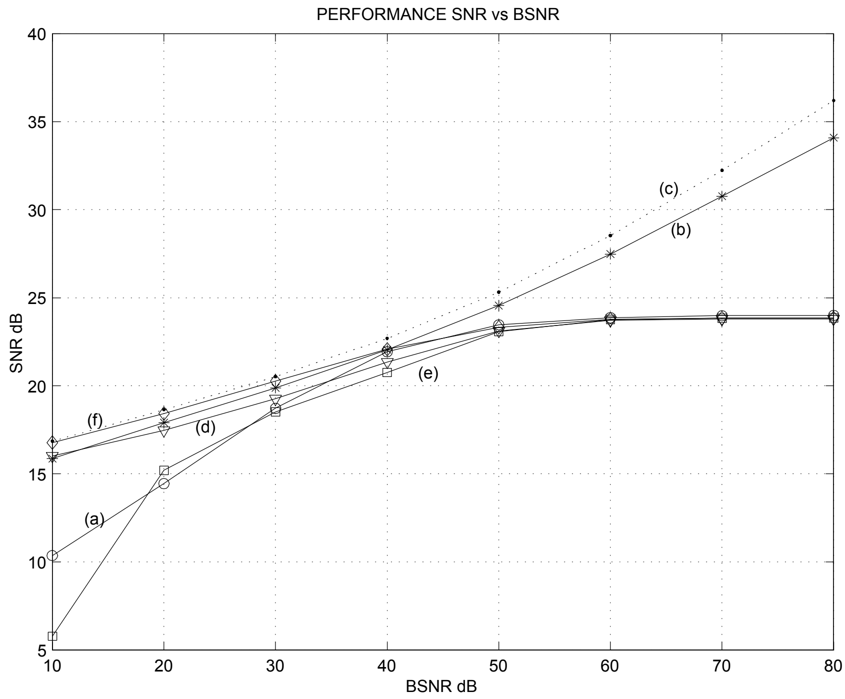


Figure 5.8: SNR of the image reconstruction versus $BSNR$. The Bar-Box blurred 9×9 PSF and the Gaussian window function LPA for deblurring kernel. The estimators and parameters: (a) RI with $\varepsilon = 0.015$ and $\Gamma = 1.6$ (curve with 'o'), (b) RWI with $\varepsilon = 0.4$ and $\Gamma = 1.6$, PSF estimated (curve with stars *), (c) RWI with $\varepsilon = 0.4$ and $\Gamma = 1.6$, PSF given (dotted curve), (d) $Matlab_1$ with $\varepsilon_1 = 0.26$ (curve with triangles); (e) $Matlab_{21}$ with $\varepsilon_2 = 55$, PSF estimated (curve with squares); (f) $Matlab_{22}$, PSF given (curve with diamonds).

Let us consider another group of experiments which are different from the discussed above only by using the Gaussian PSF for blurring and deblurring. In order to have a scenario comparable with the box-car PSF we define the parameters of the Gaussian PSF in such way that the L_2 norms of PSF in the both scenarios are equal. Then, the level of the noise in blurred and ideally deblurred image are the same for the box-car and Gaussian PSF.

This equivalent Gaussian blurring kernel have a size 15 × 15 and kernel

$$v(x_1, x_2) = \exp(-(x_1^2 + x_2^2)/0.2048),\ x_1, x_2 = -8, -7, ..0, ..., 7, 8.$$

First, we wish to determine the optimal parameters Γ^* and ε^* from the analysis of the $LPA - ICI$ deblurring algorithm by optimizing the above six performance criteria.

These performance criteria are calculated for symmetric and four quadrant window estimates as well as for the weighted final estimates (2.17). We have studied few versions of the fusion for the final estimate which are different in a window estimates used for fusion as well as the weights used in the fusion. In particular, we considered the final estimate obtained from four (quadrants 1,2,3,4 only) and from five estimates (quadrants 1,2,3,4 plus symmetric window). We considered also two types of the weights of the fusion: inverse variance of the estimates as in (2.17) and the simple mean of the fused estimates.

The analysis is again as it was above in the favor of the four quadrant estimates (symmetric window estimate is not included) with the inverse variance weights what is used further.

The criteria functions are relatively slow varying. It allows to select compromise values of Γ and ε which are nearly optimal for all criterion functions equal to selected ones for the uniform box-car PSF. These compromise values are $\Gamma^* = 1.6$ and $\varepsilon^* = 0.015$ for the RI algorithm and $\Gamma^* = 1.6$ and $\varepsilon^* = 0.4$ for the RWI algorithm.

In this scenario the behavior of the criteria as functions of Γ and ε for the symmetric and quadrant window estimates is illustrated in Figures 5.9 and 5.10 for the RI algorithm. The curves in these Figures show that the different criteria for different quadrant windowed estimates indeed can be agreed in terms of their extremum values achieved for the parameters Γ and ε having the compromise values shown above.

For the algorithm RWI the corresponding criteria curves are shown in Figures 5.11 and 5.12.

Table 5.2 similar to Table 5.1 presents criteria values for the case of the Gaussian PSF.

Table 5.2

Gaussian blur PSF and Gaussian window LPA with $m = 2$

Criteria	RWI $\varepsilon = 0.4$	RWI $\varepsilon = 0.4$ $\|Y(f)\|$ given	RI $\varepsilon = 0.015$	$Matlab_1$ $\varepsilon_1 = 0.26$	$Matlab_{21}$ $\varepsilon_2 = 55$	$Matlab_{22}$ $\|Y(f)\|$ given
SNR, dB	19.46	19.77	19.47	19.14	19.02	19.35
$ISNR$,	3.270	3.574	3.276	2.945	2.833	3.160
$RMSE$, dB	0.0564	0.0545	0.0564	0.0585	0.0593	0.0571
$PSNR$, dB	24.98	25.28	24.98	24.64	24.54	24.86
MAE	0.0302	0.0297	0.0301	0.0329	0.0325	0.0327
$MAX - DIFF$	0.5903	0.5659	0.5979	0.6051	0.6091	0.5854

The criteria based comparison of the algorithms results in conclusions which are quite similar to those we had for the box-car PSF. Again it is in favor to the RWI algorithm both in the criteria figures as well as visually.

One of the goals of the experiment with the Gaussian PSF is to demonstrate that the deconvolution of the Gaussian PSF is a more difficult problem than it is for the box-car PSF. The comparison of the criteria values in Table 5.1 versus Table 5.2 shows that the all criteria values are worse for the Gaussian PSF. It confirms also the theoretical results of Propositions 8 and 9 that the deconvolution of the super-smooth kernel (Gaussian) is more complex problem than the deconvolution of the smooth kernel (box-car). Further illustrations for the Gaussian PSF are presented in Figures 5.13 and 5.14.

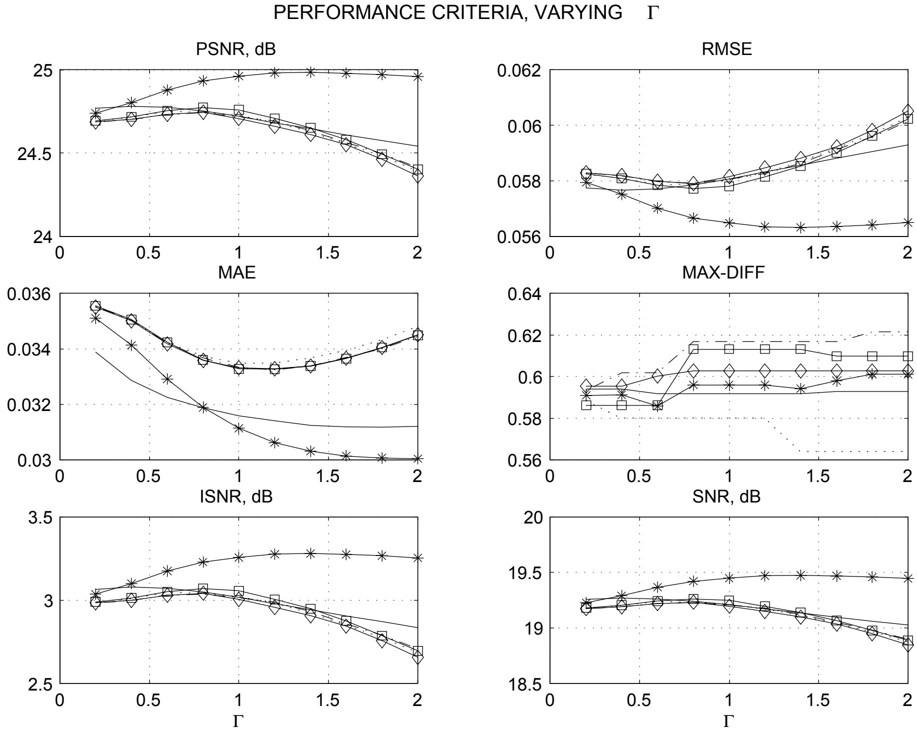


Figure 5.9: The Regularized Inverse (RI) $LPA - ICI$ estimates. The Gaussian PSF and the Gaussian window function in the LPA. The performance criteria as functions of the threshold Γ for $\varepsilon = 0.015$. Criteria are given for the symmetric, quadrant and final estimates. Notation used: solid line for the symmetric window estimate, stars * for the final fused estimate, dotted lines, dash-dot lines, lines with squares and diamonds for the Q_1, Q_2, Q_3, and Q_4 quadrant estimates respectively. Extremum values of SNR, $ISNR$, $PSNR$, $RMSE$ for the final estimate are at about $\Gamma \simeq 1.6$. Symmetric and quadrant estimates achieve their extremum values at the values close to $\Gamma \simeq 1.6$. Thus, $\Gamma \simeq 1.6$ can be treated as a compromise value for all criteria and for all estimates.

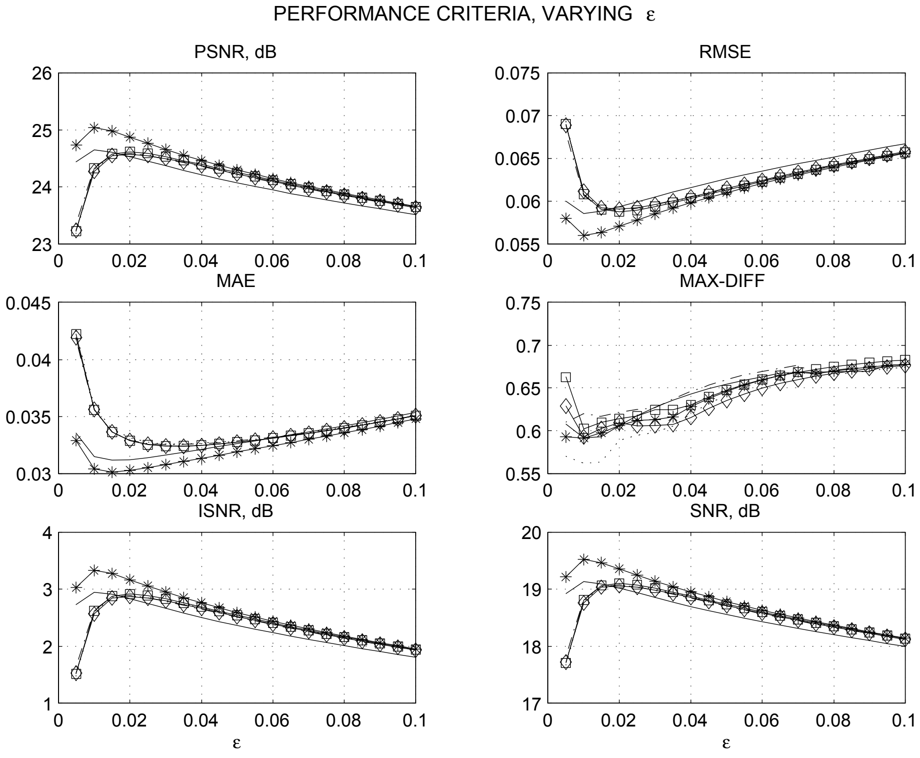


Figure 5.10: The Regularized Inverse (RI) $LPA - ICI$ estimates. The Gaussian PSF and the Gaussian window function in the LPA. The performance criteria as functions of of ε for $\Gamma = 1.6$. Criteria are given for the symmetric, quadrant and final estimates. Notation used: solid line for the symmetric window estimate, stars * for the final fused estimate, dotted lines, dash-dot lines, lines with squares and diamonds for the Q_1, Q_2, Q_3, and Q_4 quadrant estimates respectively. Extremum values of SNR, $ISNR$, $PSNR$, $RMSE$ for the final estimate are at about $\varepsilon \simeq 0.015$. Symmetric and quadrant estimates achieve their extremum values at the values close to $\varepsilon \simeq 0.015$. Thus, $\varepsilon \simeq 0.015$ can be treated as a compromise value for all criteria and for all estimates.

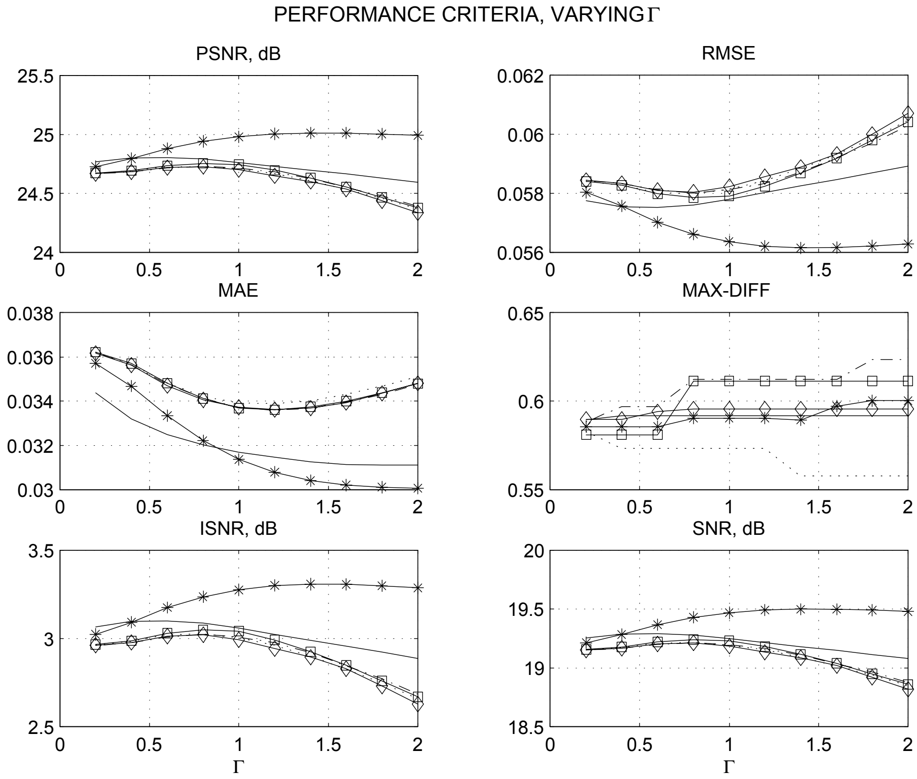


Figure 5.11: The Regularized Wiener Inverse (RWI) $LPA-ICI$ estimates. The Gaussian PSF and Gaussian window of the LPA. The performance criteria as functions of the threshold Γ for $\varepsilon = 0.4$. Criteria are given for the symmetric, quadrant and final estimates. Notation used: solid line for the symmetric window estimate, stars * for the final fused estimate, dotted lines, dash-dot lines, lines with squares and diamonds for the Q_1, Q_2, Q_3, and Q_4 quadrant estimates respectively. Extremum values of SNR, $ISNR$, $PSNR$, $RMSE$ for the final estimate are at about $\Gamma \simeq 1.6$. Symmetric and quadrant estimates achieve their extremum values at the values close to $\Gamma \simeq 1.6$. Thus, $\Gamma \simeq 1.6$ can be treated as a compromise value for all criteria and for all estimates.

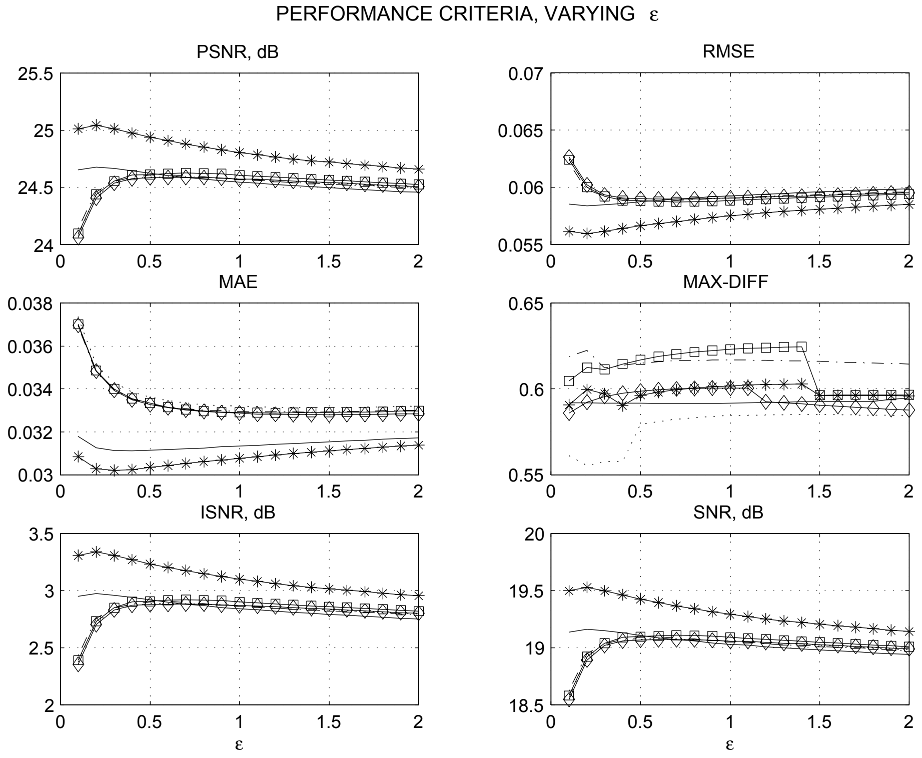


Figure 5.12: The Regularized Wiener Inverse (RWI) $LPA-ICI$ estimates. The Gaussian PSF and the Gaussian window function in the LPA. The performance criteria as functions of of ε for $\Gamma = 1.6$. Criteria are given for the symmetric, quadrant and final estimates. Notation used: solid line for the symmetric window estimate, stars * for the final fused estimate, dotted lines, dash-dot lines, lines with squares and diamonds for the Q_1, Q_2, Q_3, and Q_4 quadrant estimates respectively. Extremum values of SNR, $ISNR$, $PSNR$, $RMSE$ for the final estimate are at about $\varepsilon \simeq 0.4$. Symmetric and quadrant estimates achieve their extremum values at the values close to $\varepsilon \simeq 0.4$. Thus, $\varepsilon \simeq 0.4$ can be treated as a compromise value for all criteria and for all estimates.

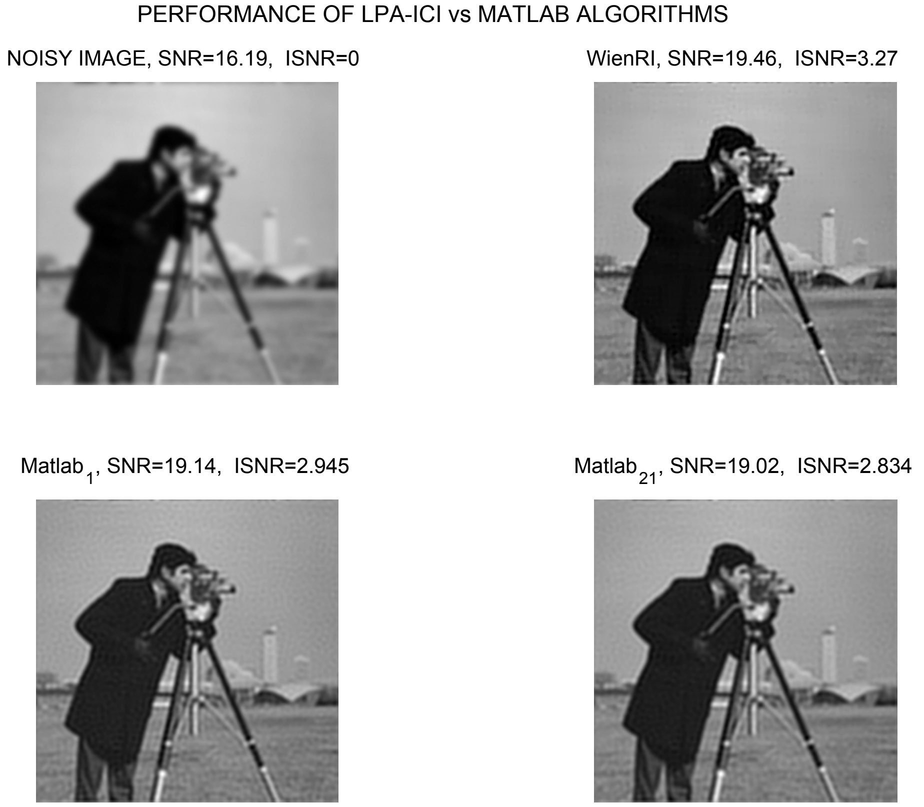

PERFORMANCE OF LPA-ICI vs MATLAB ALGORITHMS
NOISY IMAGE, SNR=16.19, ISNR=0
WienRI, SNR=19.46, ISNR=3.27
Matlab$_1$, SNR=19.14, ISNR=2.945
Matlab$_{21}$, SNR=19.02, ISNR=2.834

Figure 5.13: The RWI algorithm, Gaussian PSF, Gaussian window in the LPA. (a) The observed blurred noisy Cameraman image; (b) the final $LPA - ICI$ estimate fused from the four quadrant estimates. The estimate obtained by using the Wiener Inversion (RWI) demonstrates a performance with $SNR = 22$ dB and $ISNR = 6.85$ dB; (c) The estimate $Matlab_1$ obtain by the MATLAB routine deconvreg.m with optimization on regularization parameter with $SNR = 21.35$ dB and $ISNR = 6.16$ dB; (d) The estimate $Matlab_{21}$ obtain by the MATLAB routine deconvwnr.m with optimization on some regularization parameter with $SNR = 20.75$ dB and $ISNR = 5.57$ dB. The criteria values and the visual inspection of the image reconstructions are definitely in favor of the $LPA - ICI$ based RWI algorithm.

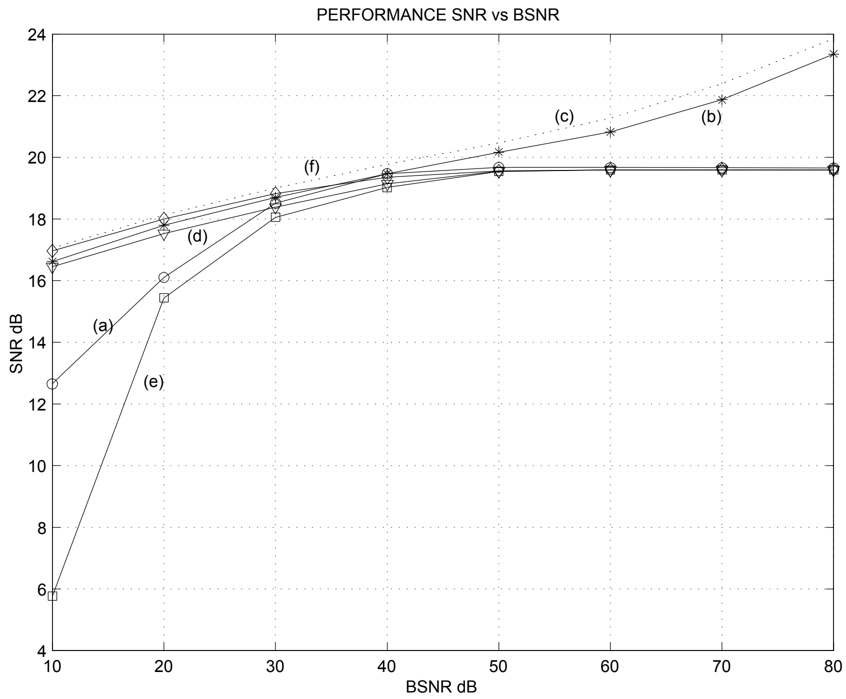


Figure 5.14: SNR of the image reconstruction versus $BSNR$. The Gaussian 15×15 PSF and the Gaussian window function LPA for deblurring kernel. The estimators and parameters: (a) RI with $\varepsilon = 0.015$ and $\Gamma = 1.6$ (curve with 'o'), (b) RWI with $\varepsilon = 0.4$ and $\Gamma = 1.6$, PSF estimated (curve with stars *), (c) RWI with $\varepsilon = 0.4$ and $\Gamma = 1.6$, PSF given (dotted curve), (d) $Matlab_1$ with $\varepsilon_1 = 0.26$ (curve with triangles); (e) $Matlab_{21}$ with $\varepsilon_2 = 55$, PSF estimated (curve with squares); (f) $Matlab_{22}$, PSF given (curve with diamonds).

5.6 Proof of propositions

5.6.1 Proposition 7

(1) The first part of the proposition follows from Proposition 1. Let us outline this derivation. Substituting the weight g_h from (1.29) in the convolution for $\hat{y}_h(x)$ we obtain

$$\hat{y}_h(x) = \sum_l g_h(\Delta(x-l))y(\Delta l), \tag{5.66}$$

$$g_h(\Delta x) = w_h(\Delta x)\phi^T(0)\Phi_h^{-1}\phi(\Delta x), \tag{5.67}$$
$$\Phi_h = \sum_x w_h(\Delta x)\phi(\Delta x)\phi^T(\Delta x).$$

Note, that the infinite $2D$ lattice is used here, the sampling intervals are equal to Δ on both variables $x_1$and x_2, $x = (x_1, x_2)$, $l = (l_1, l_2)$ and $\sum_l = \sum_{l_1}\sum_{l_2}$.

First consider the matrix Φ_h:

$$\begin{aligned} \Phi_h &= T_d(h)\frac{1}{h^2}\sum_x w(\frac{\Delta x}{h})\phi(\frac{\Delta x}{h})\phi^T(\frac{\Delta x}{h})T_d(h) = \\ &\frac{1}{\Delta^2}T_d(h)\sum_{x\in X_\infty} w(\frac{\Delta x}{h})\phi(\frac{\Delta x}{h})\phi^T(\frac{\Delta x}{h})(\frac{\Delta}{h})^2 T_d(h) \\ \rightarrow &\frac{1}{\Delta^2}T_d(h)\Phi T_d(h) \end{aligned}$$

where $T_d(h) = diag\{\phi(h)\}$ is a diagonal matrix composed from the elements of the vector $\phi(h)$ (see 1.46).

Inserting the limit expression for Φ_h in (5.67) we obtain

$$\begin{aligned} g_h(\Delta x) &\rightarrow \frac{\Delta^2}{h^2}w(\frac{\Delta x}{h})\phi^T(0)T_d^{-1}(h)\Phi^{-1}T_d^{-1}(h)\phi(\Delta x) = \\ &\frac{\Delta^2}{h^2}w(\frac{\Delta x}{h})\phi^T(0)\Phi^{-1}\phi(\frac{\Delta x}{h}) = \frac{\Delta^2}{h^2}g(\frac{\Delta x}{h}), \end{aligned} \tag{5.68}$$

where

$$g(x) = w(x)\phi^T(0)\Phi^{-1}\phi(x). \tag{5.69}$$

Further, substituting this formula in (5.66) we have

$$\begin{aligned} &\hat{y}(x,h) \rightarrow \frac{\Delta^2}{h^2}\sum_l g(\frac{\Delta(x-l)}{h})y(\Delta l) \rightarrow \\ &\frac{1}{h^2}\int_{R^2} g(\frac{x-u}{h})y(u)du = \int_{R^2} g(u)y(x-hu)du. \end{aligned}$$

It proves (5.32).

(2) For DFT of g_h we have

$$G_h(f) = \sum_x g_h(\Delta x) \exp(-i2\pi < f, x >)$$

and substituting (5.68) in it we obtain

$$G_h(f) \rightarrow \frac{1}{h^2} \sum_{x \in X_\infty} \Delta^2 g(\frac{\Delta x}{h}) \exp(-i2\pi < \frac{1}{\Delta} f, \Delta x >) \rightarrow$$
$$\frac{1}{h^2} \int_{R^2} g(\frac{x}{h}) \exp(-i2\pi < \lambda, x >) dx =$$
$$\int_{R^2} g(x) \exp(-i2\pi < \lambda h, x >) dx = G(\lambda h).$$

It proves (5.34).

(3) Using (5.68) we obtain for the squared norm of $g_h(x)$

$$\sum_x g_h^2(x) \rightarrow \frac{\Delta^2}{h^2} \sum_x \frac{\Delta^2}{h^2} g^2(\frac{\Delta x}{h}) \rightarrow \frac{\Delta^2}{h^2} \int_{R^2} g^2(u) du = \Delta^2 \int_{R^2} |G(\lambda h)|^2 d\lambda.$$

(4) According to (5.34) $G_h(f) \rightarrow G_h(\lambda)$ and $V(f) \rightarrow V(\lambda)$. Then the formula (5.20) for the variance of PI gives

$$\sigma^2_{\hat{y}_h} = \frac{\sigma^2}{n_1 n_2} || \frac{G_h(f)}{V(f)} ||_2^2 = \sigma^2 \sum_f | \frac{G_h(f)}{V(f)} |^2 \frac{1}{n_1 n_2} \rightarrow$$
$$\sigma^2 \int_{R^2} | \frac{G_h(f)}{V(f)} |^2 df = \sigma^2 \Delta^2 \int_{R^2} | \frac{G_h(\lambda)}{V(\lambda)} |^2 d\lambda,$$
$$\lambda = f / \Delta.$$

It proves (5.36). In a similar way the formulas (5.37)-(5.38) can be verified.

It completes the proof of Proposition 7.

5.6.2 Proposition 8

(I) For the analysis of the bias we use the Taylor series of $y(x-u)$ with the remainder term in the Lagrange form (1.155):

$$y(x-u)=\sum_{s=0}^{M}\sum_{|r|=s}\frac{(-1)^{|r|}}{r!}u^{r}D^{(r)}y(x)+ \qquad (5.70)$$
$$\sum_{|r|=M+1}\frac{1}{r!}u^{r}D^{(r)}y(x-\lambda u),$$
$$0\leq\lambda\leq 1.$$

Then, according to Proposition 7 the estimation bias is defined as

$$E\{e_{y}\}=y(x)-\hat{y}_{h}(x)=y(x)-\int_{R^{2}}g(u)y(x-hu)du. \qquad (5.71)$$

Substituting the series (5.70) in (5.71) with $M=\min\{m+1,r\}$ and taking into consideration the vanishing moments properties of the $m-th$ order estimator we arrive to the equality for the estimation bias

$$E\{e_{y}\}=-h^{M}\sum_{|r|=M}\frac{1}{r!}\int_{R^{2}}|g(u)u^{r}D^{(r)}y(x-\lambda hu)|du. \qquad (5.72)$$

The inequality (5.45) immediately follows from (5.72). Note that $M=\min\{m+1,r\}$ defines the maximum power of the Taylor series which can be used for the estimation error approximation.

(II) According to Proposition 7 the asymptotic variance of estimation by the PI method is given by the formula (5.36)

$$var\{e_{y}^{0}\}\leq\Delta^{2}\sigma^{2}\int_{R^{2}}|\frac{G_{h}(\lambda)}{V(\lambda)}|^{2}d\lambda. \qquad (5.73)$$

Substituting (5.41) in (5.73) we have

$$\int_{R^{2}}|\frac{G_{h}(\lambda)}{V(\lambda)}|^{2}d\lambda= \qquad (5.74)$$
$$\int_{||\lambda||\leq A}|\frac{G_{h}(\lambda)}{V(\lambda)}|^{2}d\lambda+\int_{||\lambda||>A}|\frac{G_{h}(\lambda)}{V(\lambda)}|^{2}d\lambda\leq$$
$$\frac{1}{c_{1}^{2}}\int_{||\lambda||\leq A}|G(\lambda h)|^{2}d\lambda+\frac{1}{c_{0}^{2}}\int_{||\lambda||>A}|G(\lambda h)|^{2}||\lambda||^{2\alpha}d\lambda\leq$$
$$\frac{1}{c_{1}^{2}h^{2}}\int_{R^{2}}|G(\lambda)|^{2}d\lambda+\frac{1}{c_{0}^{2}h^{2\alpha+2}}\int_{R^{2}}|G(\lambda)|^{2}||\lambda||^{2\alpha}d\lambda=$$
$$\frac{B_{0}^{2}}{c_{1}^{2}h^{2}}+\frac{B_{1}^{2}}{c_{0}^{2}h^{2\alpha+2}}. \qquad (5.75)$$

It proves (5.48) and completes the proof of Proposition 8.

5.6.3 Proposition 9

(I) The analysis of the bias is completely the same as it is in the proof of Preposition 8.

(II) The variance of the estimate is obtained by modification of the corresponding proof of Preposition 8. Substituting (5.42) in the formula for the variance (5.36), and using (5.75) we have

$$\int_{R^2} |\frac{G(h\lambda)}{V(\lambda)}|^2 d\lambda = \tag{5.76}$$

$$\int_{||\lambda||\leq A} |\frac{G(h\lambda)}{V(\lambda)}|^2 d\lambda + \int_{||\lambda||>A} |\frac{G(h\lambda)}{V(\lambda)}|^2 d\lambda \leq$$

$$\frac{1}{c_1^2}\int_{||\lambda||\leq A} |G(\lambda h)|^2 d\lambda + \frac{1}{c_0^2}\int_{||\lambda||>A} |G(\lambda h)|^2 \exp(2\gamma||\lambda||^\alpha) d\lambda \leq$$

$$\frac{B_0^2}{c_1^2 h^2} + \frac{1}{c_0^2 h^2}\int_{||u||>Ah} |G(u)|^2 \exp(2\gamma||u||^\alpha h^{-\alpha}) du =$$

$$\frac{B_0}{c_1^2 h^2} + \left\{ \begin{array}{c} \frac{1}{c_0^2 h^2}\int_{||u||>Ah} |G(u)|^2 \exp(2\gamma||u||^\alpha h^{-\alpha}) du, \text{ if } Ah \leq \bar{\Lambda}, \\ 0, \ Ah > \bar{\Lambda} \end{array} \right. \qquad \leq$$

$$\frac{B_0}{c_1^2 h^2} + \left\{ \begin{array}{c} \frac{1}{c_0^2 h^2}\exp(2\gamma\bar{\Lambda}^\alpha h^{-\alpha}) \int_{R^2} |G(u)|^2 du, \\ 0, \ Ah > \bar{\Lambda} \end{array} \right. \qquad =$$

$$\frac{B_0}{c_1^2 h^2} + \left\{ \begin{array}{c} \frac{B_0^2}{c_0^2 h^2}\exp(2\gamma\bar{\Lambda}^\alpha h^{-\alpha}), \text{ if } Ah \leq \bar{\Lambda}, \\ 0, \ Ah > \bar{\Lambda} \end{array} \right. . \tag{5.77}$$

It proves (5.57).

(III) Let us assume that $h = c(\log\frac{1}{\Delta^2})^{-1/\alpha}$ and substitute it in (5.77). It gives as $\Delta \to 0$

$$\frac{1}{\sigma^2} var\{e_y^0\} \leq \frac{\Delta^2 B_0}{c_1^2 h^2} + \frac{\Delta^2 B_0^2}{c_0^2 h^2}\exp(2\gamma\bar{\Lambda}^\alpha h^{-\alpha}) \simeq$$

$$\frac{\Delta^2 B_0^2}{c_0^2 h^2}\exp(2\gamma\bar{\Lambda}^\alpha h^{-\alpha}) = \frac{\Delta^2 B_0^2}{c_0^2 h^2}(\frac{1}{\Delta^2})^{2\gamma(\bar{\Lambda}/c)^\alpha} =$$

$$\frac{B_0^2}{c_0^2 h^2}(\Delta^2)^{1-2\gamma(\bar{\Lambda}/c)^\alpha} = \frac{B_0^2}{c_0^2 c^2}(\log(\frac{1}{\Delta^2}))^{2/\alpha}\frac{1}{(\frac{1}{\Delta^2})^{1-2\gamma(\bar{\Lambda}/c)^\alpha}}.$$

Compare the order of the last expression with the order of the bias term in the mean square error. For the squared bias we have the order $0(h^{2M}) = 0((\log\frac{1}{\Delta^2})^{-2M/\alpha})$.

For the ration of the variance to the squared bias we have

$$(\log(\tfrac{1}{\Delta^2}))^{2/\alpha}\frac{1}{(\frac{1}{\Delta^2})^{1-2\gamma(\bar{\Lambda}/c)^{\alpha}}}\frac{1}{(\log\frac{1}{\Delta^2})^{-2M/\alpha}} = \frac{(\log(\frac{1}{\Delta^2}))^{(2+2M)/\alpha}}{(\frac{1}{\Delta^2})^{1-2\gamma(\bar{\Lambda}/c)^{\alpha}}} \to 0$$
$$\text{as } \Delta \to 0.$$

It is used here that $1-2\gamma(\bar{\Lambda}/c)^{\alpha} > 0$ and that $(\log n)^a/n \to 0$ as $n \to \infty$ for $a \geq 0$.

Then we can conclude that the bias defines the main term of the mean squared error.

It proves (5.58) and completes the proof of Proposition 9.

Bibliography

[1] Abramovich, F. and B. Silverman, ”Wavelet decomposition approaches to statistical inverse problems,” *Biometrika,* vol. 85, pp. 115-129, 1998.

[2] Arce, G. R. and R. E. Foster, “Detail preserving ranked-order based filters for image processing,” *IEEE Trans. Acoustic., Speech, Signal Processing*, vol. 37, pp. 83–98, 1989.

[3] Astola J. and P. Kuosmanen, *Fundamentals of Nonlinear Digital Filtering*, New York, CRC Press, 1997.

[4] Banham M. R. and A.K. Katsaggelos, ” Spatially adaptive wavelet-based multiscale image restoration,” *IEEE Trans. Image Processing*, vol. 5, pp. 619-634, 1996.

[5] Barron, A., Birg´e and Massart, P., ”Risk bounds for model selection via penalization,” *Prob. Theory and Rel. Fields,* vol. 113, no. 3, pp. 301–413, 1999.

[6] Bertero, M., De Mol, C., and Pike, E. R. , ”Linear inverse problems with discrete data I: General formulation and singular system analysis,” *Inverse Problems*, vol. 1, pp. 301-330, 1985.

[7] Blake, A. and A. Zisserman, *Visual reconstruction,* MIT Press, Cambridge, MA, 1987.

[8] Candès E. J. and D. L. Donoho, ” Recovering edges in ill-posed inverse problems: optimality of curvelet frames,” *Annals of Statistics*, vol. 30, no. 3, pp. 784-842, 2002.

[9] Cavalier L., and Tsybakov A., ”Sharp adaptation for inverse problems with random noise,” *Preprint, PMA 559, University Paris 6*, 2000.

[10] Charbonnier, P., Blanc-Feraud, L., Aubert, G., and Barlaud, M., ” Deterministic edge-preserving regularization in computed imaging,” *IEEE Trans. on Image Processing*, vol. 6, pp. 298-311, 1997.

[11] Carasso A.S., "Image restoration and diffusion processes," *SPIE Proceedings*, 2035, pp. 255-266, 1993.

[12] Carasso A.S., "Error bounds in nonsmooth image deblurring," *SIAM J. Math Analysis,* vol. 28, pp. 656-668, 1997.

[13] Carasso A.S., "Direct blind deconvolution," *SIAM J. Appl. Math.,* vol. 61, no. 6, pp. 1980-2007, 2001.

[14] Cleveland W.S. and Loader C., "Smoothing by local regression: principles and methods". In *Statistical theory and computational aspects of smoothing* (Ed. W. Hardel and M. Schimek), pp. 10-49, Physica-Verlag, 1996.

[15] Donoho, D.L. and I.M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, 1994.

[16] Donoho, D. L.,"Wedgelets: nearly minimax estimation of edges," *Annals of Statistics*, vol. 27, pp. 859-897, 1999.

[17] Donoho, D. L., "Non-linear solution of linear inverse problems by wavelet-vaguelette decomposition," *Appl. Comput. Harm. Anal.*, vol. 2, pp. 101-126, 1995.

[18] Egiazarian, E., V. Katkovnik and J. L. Medina, "Adaptive varying window size image de-noising using the ICI rule with median filters", *SPIE Symposium, Electronic Imaging Science and Technology*, pp. 20-25, 2002, San Jose, California, USA

[19] Engel, J., "A simple wavelet approach to nonparametric regression from recursive partitioning schemes," *J. Multivariate Anal.,* vol. 49, pp. 242-254, 1994.

[20] Fan J. and Gijbels I., *Local polynomial modelling and its application.* London: Chapman and Hall, 1996.

[21] Fan J., "On the optimal rates of convergence for nonparametric deconvolution problems," *Annals of Statistics*, vol. 21, no. 4, pp. 1900-1925, 1993.

[22] Fan J., and Y.K., Truong, "Nonparametric regression with errors in variables," *Annals of Statistics*, vol.19, no. 3, pp.1257-1272, 1991.

[23] Friedlander, F.G., *Introduction to the theory of distributions*, 2nd ed., Cambridge, Cambridge University Press, 1999.

[24] Gel'fand, I.M. and Shilov, G.E., *Generalized functions, vol. 1: Properties and operations*, N.Y., Harcourt Brace, 1977.

[25] Gel'fand, I.M. and Shilov, G.E., *Generalized functions, vol. 2: Spaces of fundamental and generalized functions,* N.Y., Harcourt Brace, 1977.

[26] Geman, S. and Geman, D., "Stochastic relaxation, Gibbs distribution, and Bayesian restoration of images," *IEEE Trans. on PAMI,* vol. 6, pp. 721–741, 1984.

[27] Geman, D. and Reynolds, G. "Constrained restoration and the recovery of discontinuities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, pp. 367-382, 1992.

[28] Geman, D. and Yang, C., "Nonlinear image recovery with half-quadratic regularization," *IEEE Trans. Image Processing*, vol. 4, pp. 932-946, 1995.

[29] Chu, C. K., Glad I. K., Godtliebsen, F. and Marron, J. S., "Edge preserving smoothers for image processing (with discussion)," *J. Amer. Statist. Ass.*, vol. 93, pp. 526–556, 1998.

[30] Ganesan L. and P. Bhattacharyya, " Edge detection in untextured and textured images-a common computational framework," *IEEE Trans. systems, Man, and Cybernetics, Part B: Cybernetics,* vol. 27, no. 5, pp. 823-834, 1997.

[31] Girard, D., "From template matching to optimal approximation by piecewise smooth curves." In: Curves and Surfaces in Computer Vision and Graphics, *Proc. Conf. Santa Clara,* Calif., pp. 174–182, 1990.

[32] Glasbey, C. A. and Morgan, G. W. , *Image analysis for the biological sciences,* New York, Wiley, 1995.

[33] Godtliebsen, F., Spjøtvoll, E. and Marron, J. S., "A nonlinear Gaussian filter applied to images with discontinuities," *J. Nonparametric Statistics*, vol. 8, pp. 21-43, 1997.

[34] Granovsky B.L. and H.-G. Müller, "Optimization kernel methods: a unifying variational principle," *International Statistical Review*, vol. 59, pp. 373-388, 1991.

[35] H. M. Lin and A. N. Wilson, Jr., "Median filters with adaptive length," *IEEE Trans. Circuits Syst.,* vol. 35, no. 6, pp. 675-690, 1988.

[36] H. Hwang and R. A. Haddad, "Adaptive median filters: new algorithms and results," *IEEE Trans. on Image Processing,* vol. 4, no. 4, pp. 499-502, 1995.

[37] Goldenshluger, A. and A. Nemirovski, "On spatial adaptive estimation of nonparametric regression," *Technical Report 5/94, Technion, Haifa, 1994.*

[38] Goldenshluger, A. and A. Nemirovski, "On spatial adaptive estimation of nonparametric regression," *Math. Meth. Statistics,* vol. 6, pp. 135-170, 1997.

[39] Goldenshluger, A. and A. Nemirovski, "Adaptive de-noising of signals satisfying differential inequalities," *IEEE Trans. Inf. Theory,* vol. 43, no. 3, pp.872-889, 1997.

[40] Goldenshluger, A., "On pointwise adaptive nonparametric deconvolution," *Bernoulli,* vol. 5, pp. 907-925, 1999.

[41] Hall, P. and Raimondo, M., "On global performance of approximations to smooth curves using gridded data," *Annals of Statistics*, vol. 26, no. 6, pp. 2206–2217, 1998.

[42] A. Ben Hamza, P. Luque, J. Martinez, and R. Roman, "Removing noise and preserving details with relaxed median filters," *J. Math. Imag. Vision*, vol. 11, no. 2, pp. 161–177, 1999.

[43] Haralick, R.M., "Edge and region analysis for digital image data," *Comput. Graphics and Image Processing,* vol. 12, pp. 60–73, 1980.

[44] Haralick, R.M., "Digital step edges from zero crossing of second directional derivatives," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 58-68, 1984.

[45] R. C. Hardie and K. E. Barner, "Rank conditioned rank selection filters for signal restoration," *IEEE Trans. Image Processing*, vol. 3, pp.192–206, 1994.

[46] Hardle W. *Applied nonparametric regression.* Cambridge, University Press, Cambridge, 1990.

[47] Hastie T. and Loader C. "Local regression: automatic kernel carpentry" (with discussion), *Statistical Science*, vol. 8, no. 2, pp. 120-143, 1993.

[48] Hillery A.D. and R.T. Chin, "Iterative Wiener filters for image restoration," *IEEE Trans. Signal Processing*, vol. 39, pp. 1892-1899, 1991.

[49] Huang, J. S. and Tseng, D. H.," Statistical theory of edge detection," *Graphics and Image Processing,* 43, pp. 337–346, 1988.

[50] Huber P.J., *Robust statistics*, John Wiley &Sons Inc., 1981.

[51] Hwang H. and Haddad R.A., "Multilevel nonlinear filters for edge detection and noise suppression," *IEEE Trans. on Signal Processing*, vol.42, no. 2, pp. 249-258, 1994.

[52] Hwang H. and Haddad R.A., "Adaptive median filters: new algorithms and results ," *IEEE Trans. on Image Processing*, vol. 44, pp. 499 -502, 1995.

[53] Hurvich C. M. and Simonoff J. S., "Smoothing parameter selection in nonparametric regression using an improved AIC criterion," *Journal of the Royal Statistical Society*, Ser. B, vol. 60, pp. 271-293, 1998.

[54] Johnstone I.M. , "Wavelet shrinkage for correlated data and inverse problems: adaptivity results," *Statistica Sinica*, vol. 9, pp. 51-83, 1999.

[55] Katkovnik V., "Problem of approximating functions of many variables," *Autom. Remote Control*, vol. 32, no. 2, Part 2, pp. 336-341, 1971.

[56] Katkovnik V., "Homogeneous integral averaging operators obtained by the method of least squares," *Autom. Remote Control*, vol. 32, no. 11, Part 1, pp. 1767-1775, 1971.

[57] Katkovnik V., "Linear and nonlinear methods of nonparametric regression analysis," *Soviet Journal of Autom. and Inform. Sciences*, vol. 5, pp. 25-34, 1979.

[58] Katkovnik V., *Linear estimation and stochastic optimization problems.* Nauka, Moscow, 1976 (in Russian).

[59] Katkovnik V. *Nonparametric identification and smoothing of data (Local approximation methods).* Nauka, Moscow, 1985 (in Russian).

[60] Katkovnik V., "A new method for varying adaptive bandwidth selection," *IEEE Trans. on Signal Processing,* vol. 47, no. 9, pp. 2567-2571, 1999.

[61] Katkovnik V. , K. Egiazarian, and J. Astola, "Median filter with varying bandwidth adaptive to unknown smoothness of the signal", *Int. Conference on Circuits and Systems (ISCAS'2000),* May 28-31, 2000, Geneva, Switzerland, Proceedings of ISCAS'2000, vol. , pp. 519 -522.

[62] Katkovnik V. , K. Egiazarian, and J. Astola, "Local transform-based image de-noising with adaptive window size selection," *EOS/SPIE Symposium, Image and Signal Processing for Remote Sensing,* September 25-29, 2000, Barcelona, Spain.

[63] V. Katkovnik, H. Oktem, K. Egiazarian, "Filtering heavy noised images using the ICI rule for adaptive varying bandwidth selection," *Proc. of Int. Conf. on Circuits and Systems (ISCAS'1999)*, vol. 4, pp. 195 -198, 1999.

[64] Katkovnik V., Egiazarian K. and J. Astola, "Median filter with varying bandwidth adaptive to unknown smoothness of the Signal", *Proc. of IEEE Int. Symposium on Circuits and Systems (ISCAS 2000),* Geneva, May 28-31, 2000, Switzerland.

[65] Katkovnik V., K. Egiazarian, and Jaakko Astola, "Weighted median filter with varying adaptive window size," *Proc. of the IASTED International Conference on Signal Processing and Communications, (ISCAS'2000)*, pp. 329-323, 2000, Marbella, Spain.

[66] Katkovnik V., K. Egiazarian, and Jaakko Astola, "Local transform-based image de-noising with adaptive window size selection," *Proc. of the EUROPTO International Conference on Image and Signal Processing for Remote Sensing (EOS/SPIE Symposium on Remote Sensing)*, 2000, Barcelona, Spain.

[67] Katkovnik V., K. Egiazarian, and Jaakko Astola, "Adaptive window size image de-noising based on intersection of confidence intervals (ICI) rule," *Journal of Mathematical Imaging and Vision,* vol. 16, pp. *223-235, 2002.*

[68] Egiazarian K., V. Katkovnik, H. Oktem and J. Astola, "Transform-based denoising with parameters adaptive to unknown smoothness of the signal". In Ed. Creutzburg and Egiazarian: *Spectral Techniques and Logic Design for Future Digital Systems.* Proc. of Int. Workshop SPECLOG'2000, Tampere, TTKK, Monistamo, Finland.

[69] Katkovnik V., K. Egiazarian, and I. Shmulevich, "Adaptive varying window size filtering based on the intersection of confidence intervals rule," *IEEE - EURASIP Workshop on Nonlinear Signal and Image Processing*, 2001, Baltimore, Maryland, USA.

[70] Katkovnik V., K. Egiazarian, and Jaakko Astola, "Application of the ICI principle to window size adaptive median filtering," *Signal Processing,* vol. 83, no. 2, pp. 251 - 257, 2003.

[71] Katkovnik V., A. Gershman, and LJ. Stankovic, "Sensor array signal tracking using a data-driven window approach," *Signal Processing*, vol. 80, N12, pp. 1507-2515, 2000.

[72] Katkovnik V. and LJ. Stankovic, "Periodogram with varying and data-driven window length," *Signal Processing*, vol. 67, N3, 1998, pp. 345-358.

[73] Katkovnik V. "On adaptive local polynomial approximation with varying bandwidth", *Proc. of IEEE Int. Conf. of Acoustics, Speech & Signal Processing (ICASSP'98),* vol.4, pp. 2321-2324, Seattle, Washington, 1998.

[74] Klemelä J. and A. Tsybakov, "Sharp adaptive estimation of linear functionals," *Annals of Statistics,* vol. 29, no. 6, pp.1537-1566, 2001.

[75] Korostelev, A. and A. Tsybakov, *Minimax theory of image reconstruction.* Springer Verlag, New York–Heidelberg–Berlin, 1993.

[76] Lancaster P. and K. Saulcauskas, "Surfaces generated by moving least squares methods, "*Mathematics of Computation*, vol. 37, no. 155, pp 141-158, 1981.

[77] Lee, J.S., "Digital image smoothing and the sigma-filter," *Computer Vision, Graphics and Image Processing,* vol. 24, pp. 255–269, 1983.

[78] Lepski, O.V. , "One problem of adaptive estimation in Gaussian white noise," *Theory Probab. Appl.*, vol. 35, no. 3, pp. 459–470, 1990.

[79] Lepskii, O. ,"Asymptotically minimax adaptive estimation I: Upper bounds. Optimally adaptive estimates," *Theory Probab. Appl.,* vol. 36, pp. 682-697, 1991.

[80] Lepski, O.V. " Asymptotic minimax adaptive estimation. 2. Statistical model without optimal adaptation. Adaptive estimators," *Theory Probab. Appl.*, vol. 37, no. 3, pp. 468–481, 1992.

[81] Lepski, O., Mammen, E. and Spokoiny, V. ," Ideal spatial adaptation to inhomogeneous smoothness: an approach based on kernel estimates with variable bandwidth selection," *Annals of Statistic*s, vol. 25, no. 3, 929–947, 1997.

[82] Lepski, O. and Spokoiny, V. ," Optimal pointwise adaptive methods in nonparametric estimation," *Annals of Statistics,* vol. 25, no. 6, pp. 2512–2546, 1997.

[83] Loader C., *Local regression and likelihood,* Series Statistics and Computing, Springer-Verlag New York, 1999.

[84] Mallat S., *A Wavelet tour of signal processing,* Academic Press, 1999.

[85] McDonald J.A. and Owen A.B., "Smoothing with split linear fits," *Technometrics,* vol. 28, no. 3, pp. 195-208, 1986.

[86] Müller, H.G. and Song, K.S., "Maximum estimation of multidimensional boundaries," *J. Multivariate Anal.*, vol. 50, no. 2, pp. 265–281, 1994.

[87] Mumford, D. and Shah, "Optimal approximation by piecewise smooth functions and associated variational problem," *Comm. Pure Appl. Math.*, vol. 42, pp. 577–685, 1989.

[88] Nadaray, E.A., "On estimating regression, "*Theory Prob. Appl.,* vol. 9, pp. 141-142, 1964.

[89] I. Pitas, A. N. Venetsanopoulos, *Nonlinear digital filters: principles and applications*, Kluwer Academic Publishers, 1990.

[90] K. N. Platoniotis, A. N. Venetsanopoulos, *Color image processing and applications*, Springer, 2000.

[91] Schucany W.R., "Adaptive bandwidth choice for kernel regression," *JASA*, vol. 90 (430), pp. 535-540, 1995.

[92] Stewart, C. V.,"Robust parameter estimation in computer vision," *SIAM Review*, vol. 41, no. 3, 1999, pp. 513–537.

[93] Wendt P. D. , Coyle E. J. , and N. C. Gallagher, "Stack filters," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-34, pp. 898–919, 1986.

[94] Yang R., L. Yin, M. Gabbouj, J. Astola, and Y. Neuvo, "Optimal weighted median filters under structural constraints," *IEEE Trans. Signal Processing,* vol. 43, pp. 591–604, 1995.

[95] Lin H.M. and A.N. Wilson, Jr., "Median filters with adaptive length," *IEEE Trans. on Circuits and Systems*, vol. 35, no. 6, pp. 675-690, 1988.

[96] Mammen, E. and A. Tsybakov,"Asymptotical minimax recovery of sets with smooth boundaries," *Annals of Statistics,* vol. 23, no. 2, pp. 502–524, 1995.

[97] Nemirovski, A. Topics in non-parametric statistics, *Lecture notes in mathematics,* 1738*,* Springer, N.Y., 2000*,* pp*. 85-277.*

[98] Nason, G. P. and Silverman, B. W., "The discrete wavelet transform in S," *J. of Computational and Graphical Statistics*, vol. 3, pp. 163-191, 1994.

[99] Neelamani R. , H. Choi, and R. G. Baraniuk, "Wavelet-Based Deconvolution Using Optimally Regularized Inversion for Ill-Conditioned Systems," *SPIE Proceedings*, vol. 3813, pp. 58-72, 1999.

[100] R. Neelamani, H. Choi, and R. G. Baraniuk, "Wavelet-based deconvolution for ill-conditioned systems," Submitted to the *IEEE transactions on Image Processing*, 2000.

[101] O'Sullivan, F., "A statistical perspective on ill-posed inverse problems," *Statist. Sci.*, vol. 1, pp. 502-527, 1986.

[102] Parzen, E., "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 81, pp. 215-222, 1962.

[103] A. A. Pervozvanskii, "*Random processes in nonlinear control systems*, Academic Press*, 1965.*

[104] Polzehl, J. and Spokoiny, V., "Adaptive weights smoothing with applications to image restoration," *J. of Royal Stat. Soc.*, vol. 62, Series B, pp. 335–354, 2000.

[105] Polzehl, J., Spokoiny, V., "Image denoising: pointwise adaptive approach," *Annals of Statistics*, vol. 31, no. 1, 2003.

[106] C. A. Pomalaza-Raez and C. D. McGillem, "An adaptive, nonlinear edge preserving filter," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 32, pp. 571-576, 1984.

[107] Pratt, W. K., *Digital Image Processing*, Wiley, New York, 1978.

[108] Qiu, P., "Nonparametric estimation of jump surface," *Sankhya,* Ser A 59, pp. 268–294, 1997.

[109] Qiu, P.,"Discontinuous regression surface fitting," *Annals of Statistics,* vo; 26, no. 6, pp. 2218–2245, 1998.

[110] Ripley, B., *Statistical inference for spatial processes,* Cambridge Univ. Press, 1988.

[111] Rosenfeld, A. and A.C. Kak, *Digital picture processing*, Academic Press, London, 1982.

[112] Ruppert D. "Empirical-bias bandwidths for local polynomial nonparametric regression and density estimation," *Journal of American Statistical Association*, vol. 92 (439), pp. 1049-1062, 1997.

[113] Scott, D. W., *Multivariate density estimation*, New York: Wiley, 1992.

[114] Spokoiny, V., "Estimation of a function with discontinuities via local polynomial fit with an adaptive window choice," *Annals of Statistics,* vol. 26, no..4, pp. 1356–1378, 1998.

[115] Suoranta, R.; Estola, K.-P. "Robust median filter with adaptive window length, " *IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 108 -111, 1991.

[116] Terzopoulos, D., "Regularization of inverse visual problems involving discontinuities," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 413-424, 1986.

[117] Tikhonov A.N. and V.Y. Arsenin, *Solution of ill-posed problems*, New York, Wiley, 1977.

[118] Titterington, D.M., "Common structure of smoothing technique in statistics," *Intern. Stat. Review*, vol. 53, pp. 141–170, 1985.

[119] Tsybakov, A., “Robust reconstruction of functions by the local-approximation methods,” *Problems of Information Transmission*, vol. 22, 1986, pp. 133–146

[120] Tsybakov, A.,” Optimal estimation accuracy of nonsmooth images,” *Problem Inf. Trans.*, vol. 25, pp. 180–191, 1989.

[121] Tsybakov, A., "Sharp adaptive estimation of linear functionals,” *Annals of Statistics*, vol. 29, pp 1567-1600, 2001.

[122] Wahba, G. , *Spline models for observational data* , CBMS-NSF Regional Conference Series in Applied Mathematics 59. SIAM, Philadelphia, 1990.

[123] Watson, G.S., Smooth regression analysis, *Sankhya,* Ser. A, vol. 26, pp. 359-372, 1964.

[124] Winkler, G. , *Image analysis, random fields and dynamic Monte Carlo methods,* Heidelberg, Springer, 1995.